Video Recommendations Based on Visual Features Extracted with Deep Learning

Tord Kvifte
Supervisor: Assoc. Prof. Dr Mehdi Elahi

Master's Thesis
Department of Information Science and Media Studies
University of Bergen

June 1, 2021

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Dr. Mehdi Elahi, for the motivation, support, and good advice throughout the entire process of developing and writing the thesis. The level of engagement you have shown to my project, me as your student, as well as research and innovation in general, is inspirational. I would also like to thank Agnete Bech Augustinussen, my girlfriend, with whom I have shared office (the living room) during most of the project due to the Covid-19 pandemic. In addition to your support and encouragement, the daily lunch walks along the beach promenade with you have been vital to keep a clear mind in an otherwise monotone time. My thanks also go to those of my fellow students who have given me input and kept in touch despite the limited possibilities of physical meetings. I would especially like to thank my good friend Lucie Sun for the editorial support in the finishing stages of the thesis.

I am very grateful for the research grant provided by the Faculty of Social Sciences at the University of Bergen, which was vital for recruiting enough participants for the study. Additionally, this work was supported by industry partners and the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through The Centres for Research-based Innovation scheme, project number 309339.

Copenhagen, May 2021

Tord Kvifte

Abstract

When a movie is uploaded to a movie Recommender System (e.g., YouTube), the system can exploit various forms of descriptive features (e.g., tags and genre) in order to generate personalized recommendation for users. However, there are situations where the descriptive features are missing or very limited and the system may fail to include such a movie in the recommendation list, known as *Cold-start* problem. This thesis investigates recommendation based on a novel form of content features, extracted from movies, in order to generate recommendation for users. Such features represent the visual aspects of movies, based on Deep Learning models, and hence, do not require any human annotation when extracted. The proposed technique has been evaluated in both offline and online evaluations using a large dataset of movies. The online evaluation has been carried out in a evaluation framework developed for this thesis. Results from the offline and online evaluation (N=150) show that automatically extracted visual features can mitigate the cold-start problem by generating recommendation with a superior quality compared to different baselines, including recommendation based on human-annotated features. The results also point to subtitles as a high-quality future source of automatically extracted features. The visual feature dataset, named DeepCineProp13K and the subtitle dataset, CineSub3K, as well as the proposed evaluation framework are all made openly available online in a designated Github repository.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Over the past decade, a wider range of media content has become increasingly available to consumers through digital streaming services. In line with this trend, it is becoming progressively more difficult for consumers to make choices. This has led to the challenge of *Choice Overload*, where consumers have a high number of options while lacking sufficient personal experience of the alternatives needed in order to make a good decision [84, 102]. Digital video streaming services, such as YouTube, are particularly prone to choice overload. With millions of hours of video content, a human being is only able to browse through a tiny fraction of the item catalog. Recommender systems can mitigate these types of challenges for users by providing short, personally tailored lists of options that satisfy user preferences, needs, and constraints [84].

In recent years, various types of video recommendation algorithms have been proposed and evaluated, demonstrating excellent performance. These algorithms typically receive different types of input data, e.g., content-associated data, also known as *content features*, and build recommendations on top of this data [3, 28, 43, 74, 75, 112]. Such recommendation techniques, that rely on descriptive data about the content of videos, are called *Content-Based Filtering (CBF)*. While these approaches can be effective in generating relevant recommendation for users, they may fall short to recommend videos where descriptive data are missing or very limited. This type of situation is called the *New Item* problem and is part of one of the most prominent and persistent issues for recommender systems, known as *Cold Start*.

Another limitation of CBF recommendation techniques is the vulnerability to over-specialization, i.e. the lack of ability to provide diverse recommendations [56]. *Collaborative Filtering (CF)* represents another popular approach in recommender systems, utilizing rating information to generate recommendations. While CF systems generally perform well in providing diverse recommendations, the new item problem for these systems occurs when an item has yet to be rated. To mitigate the limitations of each of the CBF and CF methods and achieve better performance, characteristics of both types can be combined in a *Hybrid Recommender*

System [2, 16].

Apart from the new item problem, the process of collecting quality data to represent the videos is itself another major problem in movie recommendation based on CBF techniques. Traditionally, this type of data is dependent on manual operations. For some forms of data (e.g., genre), a group of experts is essentially required to manually annotate, while other forms (e.g., rating and tag) may need a large community of users willing to provide the data. This makes the aforementioned data to be very expensive and extremely sparse to collect [21, 23, 35, 83, 124].

Methods for utilizing new technologies for automatic extraction of features to represent multimedia content offer possibilities to mitigate the new item problem. Additionally, using such methods makes recommender systems less prone to human biases and errors [30]. Automatically extracted features from movie trailers have already been demonstrated to provide promising results in generating movie recommendations [30, 31, 33, 39, 40]. At the same time, the field of computational image recognition is having a renaissance through the use of Convolutional Neural Networks (CNN) and the tremendous progress in *Deep Learning* over the past decade [46]. In other domains, such as recommender systems in fashion, deep learning-based visual features are already highly utilized [53, 60, 78]. Research on the use of deep learning to create visual features for the purpose of video recommendation is a growing topic, where different methods are still being experimented with.

The progress in deep learning has also had a dramatic impact on computational speech recognition, which has enabled high-quality automatically produced closed-captioning of videos [46]. Providing automatically generated quality data, this can make subtitles an attractive source of content features for video recommendations. While the use of lyrics have been explored in music recommender systems [20, 48, 72, 72, 80], there has, to the author's knowledge, been little research performed on subtitles for video recommendations.

Research on recommender systems normally focus on the predictive accuracy of prediction algorithms. A limitation with predictive analysis is that accuracy only constitutes for one part of the user experience of recommender systems. User experience is also influenced by other factors, such as objective system aspects (e.g. diversity) and situational or personal aspects [62]. User-centric approaches that include real users provide a significantly more robust evaluation of how well a recommender system serves its purpose by taking these other factors into account.

1.2 Problem

Despite recent advancements in recommender systems, there are still problems that have not yet been completely resolved. Some of these problems derive from the nature of the data used by recommendation algorithms, causing issues such as cold start [105]. Other problems come from the nature of the algorithms themselves, causing issues related to diversity in recommendations and scalability [105]. The capabilities of image recognition for recommender systems in other domains have been widely demonstrated, and the use of deep learning-based visual features for movie recommendations is a growing research topic. While content features traditionally used in movie recommender systems (e.g. tags and genre) rely on manual operations, deep learning-based visual features can be extracted automatically. An automatic approach may alleviate the cold start problem when other content features are missing or sparse, in addition to being cheaper to collect than manual features. Furthermore, hybrid recommender systems that combine CBF and CF techniques can be used to achieve higher recommendation performance [2, 16]. While most research on recommender systems only focus on algorithmic performance, a user-centric approach is necessary to account for other dimensions of the user experience of recommender system performance.

This thesis addresses the cold start problem by proposing a novel hybrid movie recommendation technique with the use of deep learning-based visual features. The approach is evaluated in a predictive analysis by comparing with models receiving manual features, i.e. tag and genre. Novel content features based on subtitles are also used in the evaluation to represent a baseline that can be formed of automatically extracted features. In order to account for factors that go beyond predictive performance, a framework for evaluating movie recommender models with real users is developed. The framework is utilized to investigate how real users experience the performance of the novel hybrid recommender model implemented with deep learning-based visual features, compared with hybrid models based on subtitles and traditional manual features.

The purpose of this thesis is to investigate whether automatically extracted content features in the form of deep learning-based visual features can mitigate the new item problem for movie recommendations.

1.3 Research Questions

In order to address the different aspects of the general problem statement, the thesis attempts to answer the following research questions:

- **RQ 1:** *Can visual features automatically extracted with deep learning provide better recommendation quality compared to the other types of content features?*
  - **RQ 1.1:** *Can automatic visual features provide better recommendation quality compared with traditional features that are collected manually (e.g. tag and genre)?*
  - **RQ 1.2:** *Can automatic novel visual features provide better recommendation quality compared to the subtitles of movies?*
- **RQ 2:** *How is the quality of recommendation based on visual features perceived by users compared to the other types of content features?*

- **RQ 2.1:** *How is the user perception impacted by different recommendation techniques utilizing different recommendation algorithms?*
- **RQ 2.2:** *How do the personal characteristics of users (e.g., personality, and demographics) affect their perception of different movie recommendation techniques using different types of content features?*
- **RQ 2.3:** *What is the perceived usability of a recommender system that utilizes visual features for recommendation?*

1.4 Contributions

The main contributions of my thesis include the following items:

- Proposing a novel hybrid recommendation technique based on visual features automatically extracted with deep learning.
- A comprehensive evaluation of proposed recommendation approach in both offline and online experiments, including consideration of different optimization methods and comparisons with different baselines on various evaluation metrics.
- Extracting a large dataset with visual features from 12,875 movie trailers, using an advanced deep learning model; the dataset, named DeepCineProp13K, is published openly in the project's Github repository[1].
- Collecting a large dataset of subtitles from 3,405 full length movies and exploiting them in a baseline recommendation technique; the dataset, named CineSub3K, is made openly available in the project's Github repository.
- Developing a framework for evaluating and comparing different movie recommender models with real users as a modern web application, including an evaluation of the framework's usability; source code is made openly available in the project's Github repository.

1.5 Thesis Outline

The following items describe the general outline of the thesis:

- **Chapter 2: Background.** Describes the literature related to the research problems of this thesis: Section 2.1 gives a background on movie recommender systems; Section 2.2 describes previous work on visual features for recommender systems; Section 2.3 provides a background on video captions in relation to this thesis; Section 2.4 gives an overview of user-centric evaluation of recommender systems.

[1] https://github.com/2rd/Thesis

- **Chapter 3: Methodology.** Details the methods used in the different stages of the thesis: Section 3.1 describes how data in DeepCineProp13K and CineProp3K were extracted from movies; Section 3.2 details the aggregation and refining of the datasets; Section 3.3 provides details about the recommendation algorithms used in the experiments; Section 3.4 reports the design process of the recommender system evaluation framework; Section 3.5 defines the methodology of the experiments utilized in the evaluation of the work.

- **Chapter 4: Results.** Contains the results from the experiments performed to evaluate the proposed research approach: Section 4.1 describes the results from the exploratory analysis; Section 4.2 reports the performance of the different recommendation approaches on algorithmic performance metrics; Section 4.3 provides a comparison of algorithmic performance for the tested recommendation approaches with different loss functions; Section 4.4 gives an analysis of the results from the real-user study.

- **Chapter 5: Conclusions and Future Work.** Discusses and summarizes the findings of the Results chapter in regards to the formulated research questions, the limitations of the work, as well as suggestions for further research within the problem areas of this thesis.

Chapter 2

Background

The background chapter provides an overview over previous works relevant to this thesis and is divided into 5 sections. Section 2.1 provides a background of recommender systems in the context of movie recommendation, as well as different recommendation techniques. Section 2.2 details previous works that uses visual features for movie recommendations. Section 2.3 provides a brief introduction to subtitles as a data source for recommender systems. Section 2.4 details the considerations and frameworks for online evaluation of recommender systems. Section 2.5 provides a summary of the chapter and specifies how this thesis differs from previous works.

2.1 Movie Recommender Systems

Through the internet, there is a continually growing availability of different products and an increase of data associated with products. In line with this trend, consumers are faced with progressively more difficult choices in their daily life. This has led to the challenge of *Choice Overload*, where consumers have a high number of options while lacking sufficient personal experience of the alternatives needed in order to make a good decision [86, 103]. Online video streaming services such as YouTube are particularly prone to choice overload. With millions of hours of video content, a human being is only able to browse through a tiny fraction of the item catalog. Recommender systems can mitigate this type of challenges for users by providing short, personally tailored lists of options that satisfy user preferences, needs, and constraints [86].

There exists a number of approaches to creating personalized video recommendations for users. One of the most popular types of recommender systems is based on the Content-based Filtering (CBF) technique. In CBF, items are represented by their content and the users by associating their preferences with the item content [52, 57, 74, 77, 98]. Other popular types of recommendation systems include Collaborative Filtering (CF) and Knowledge-based recommender systems. CF systems recommend items based on previous ratings from other users with similar taste, as well as the active user's previous ratings [44, 104]. Knowledge-based recommender systems, on the other hand, take into account the user's needs and constraints to predict the utility an item constitutes for them [86, 104].

By combining recommender system techniques in a Hybrid recommender system, one technique can complement the limitations of the other, and vice versa [17]. For instance, a pure CF system will not be able to recommend items that, in a new item scenario, have yet to receive any ratings. However, if there are descriptive features available for the items, content-based techniques can be used to take advantage of these to make predictions.

In the movie domain, item content is described with a set of representative features characterizing different aspects of the movie content. Traditional examples of content features are genre and tag, representing some form of *semantics* within the movies. Figure 2.1 depicts the high-level architecture of CBF recommender systems. Recommendation is performed in a three-step process, each handled by a separate component [104]. Content data is first cleaned, engineered and then used to create a *Vector Space Model* where the video items are represented as vectors of attributes. This step is handled by the *content analyzer* which provides the input to the *profile learner* and *filtering component*. The profile learner constructs user profiles based on items that the user has liked or disliked in the past. Finally, the filtering component suggests relevant items to the active user by finding video items that share similar attributes with other items that match the user profile.

Movie content features can be divided into classes in a three-level hierarchy of low-, intermediate-, and high-level features with each class illustrating different representations of

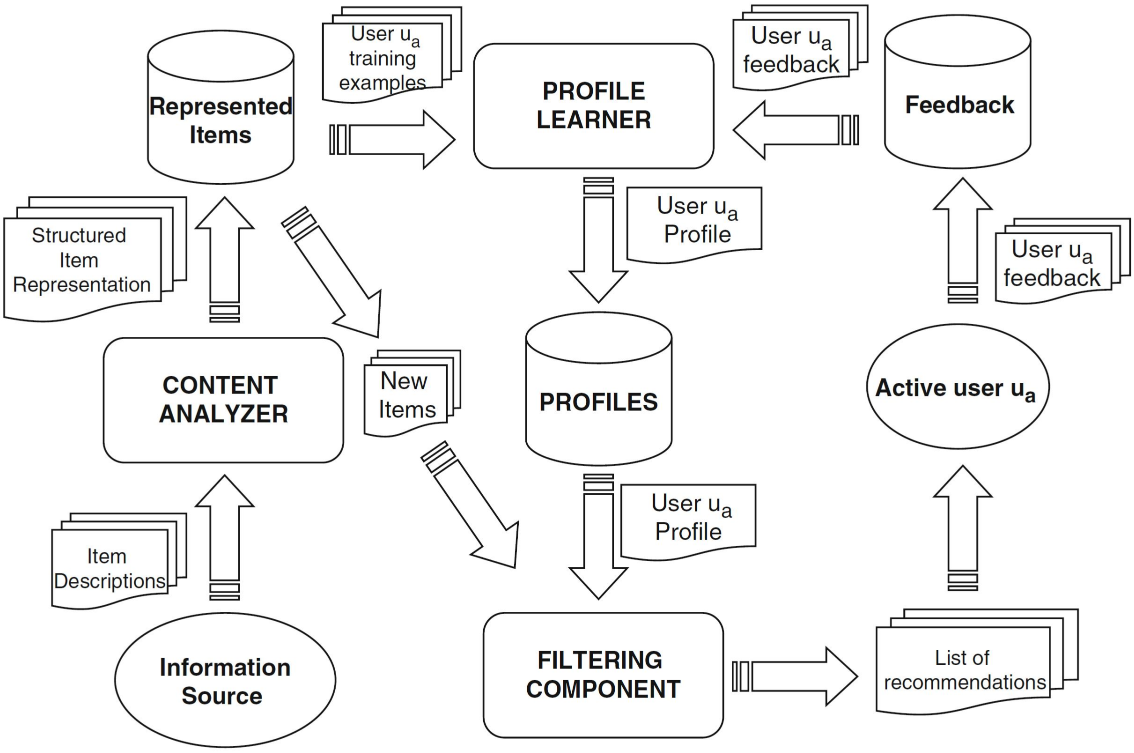


Figure 2.1: High level illustration of a content-based recommender system architecture [104]

the movies [30, 125]:

1. Representing the high-level features are the *semantic* features of a movie, dealing with events or concepts. The plot of the movie *Lawrence of Arabia*, which covers the Allies' campaign in the Middle East during World War I as seen through the eyes of T. E. Lawrence, would be an example of semantic feature.

2. Representing the intermediate-level features are the *syntactic* features of a movie, dealing with objects and their interactions. In the same noted movie, examples of syntactic features include the actor Peter O'Toole, as well as objects such as camels, horses, and daggers.

3. Representing the low-level features are the *stylistic* features of a movie, relating to the aesthetic and visual design of a movie, known as the mise-en-scéne form [31]. In the same movie as noted, examples of stylistic features include predominant colors yellow and brown, as well as long-lasting shots.

2.2 Visual Features in Movie Recommender Systems

In the domain of content-based video recommender systems, most of the prior works have been based on semantic features. These semantic features include structured data such as

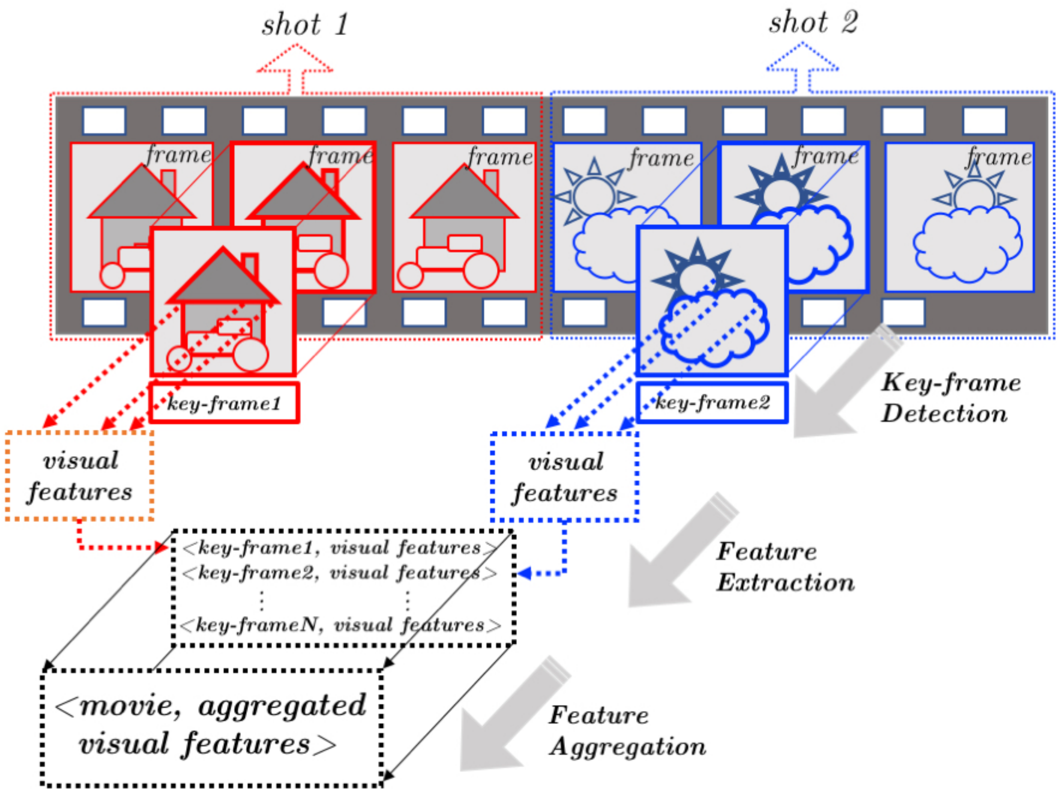


Figure 2.2: Method for extraction and aggregation of visual features from movie trailers as illustrated in Moghaddam et al. [84].

genre, cast, and director, or unstructured data, such as tags, textual reviews, and plot. In more recent works, the promise of computationally extracted low-level visual features as the basis for recommendations has been demonstrated [29, 33, 41, 84]. These features can either be used in combination with other content-based techniques or individually [32].

Visual features is a more *stylistic* approach of representing movies. This type of novel features, in contrast to the traditional features, does not need any expensive human-annotation and can be extracted automatically adopting *Computer Vision* methods. Hence, they could be a potential solution for movie recommendation in cold start, i.e., when recommending movies with no descriptive features. Another advantage of the visual features is that they can be more representative of the production style and can enable movie recommender systems to become *style-aware* [19, 68, 128, 129].

Deldjoo et al. [32] propose such a system that automatically analyzes the content of videos and extract a set of representative stylistic features. The selection of features is grounded in Applied Media Theory. Features are automatically extracted from identified key frames that are then analyzed, resulting in both temporal and spatial features such as shot length, object motion, color, and lighting. Using a conventional k-nearest neighbor algorithm on these features, the system achieves higher recommendation accuracy compared with conventional genre-based recommendations. Movie trailers also prove to be as useful in recommendations as their corresponding full-length movies when using this technique [32].

Another demonstration of the power of visual features in video recommenders is provided in Moghaddam et al. [85]. In cold start situations, when a movie recommender system is unable to provide personalized recommendations, many systems would suggest popular movies instead. While popularity is usually based on number of ratings provided by existing users, this approach may not work well when movies are new. By extracting visual features from key frames of movie trailers and aggregating these, recommender models were trained to predict the popularity and rating of movies. Their experimental results show that while there is a correlation between rating and popularity, there is also a correlation between visual features and popularity. Through the predictive analysis, they confirm that their classification model can be used to predict the success of a movie in terms of rating and popularity, even before the full movie is available [85].

Rimaz et al. [106] explore the potential of using low-level visual features in movie recommender systems. The visual features were extracted from 1800 movie trailers and combined with semantic features from corresponding movie data in the MovieLens 1M dataset. In their exploratory analysis on the visual features, they examine the evolution of visual features over time as well as investigate the visually similar clusters that could exist among movies. In their experimental evaluation where a recommender based on the extracted visual features is compared with models based on other content features such as genre, tags and a combination of these, the findings show that the model based on visual features outperforms the other models [106].

While the mentioned papers above consider low-level visual features to address the cold start problem, Li et al. [71] propose a CBF video recommender that takes advantage of deep convolutional neural networks(CNN) to extract visual features for videos. In addition to the visual modality, they also include audio and metadata features in their recommender. When comparing the performance of the three different modalities, the vision model exceeds the other models. While demonstrating the possibility of using CNN to address the cold start problem and its superiority over two other models, the study also has some significant limitations. The models are only trained on trailers for 40 TV shows with an average of 5 trailers for each TV show. There is also serious incompleteness in their test data, which is addressed by utilizing synthetic anchor points to bridge the gap between training and test data [71].

Filho, Wehrmann and Barros [42] propose a purely content-based recommender system named DeepRecVis, built on visual features extracted from keyframes of movie trailers with CNN. The CNN model is pre-trained on ImageNet and Places-365, with the purpose of letting it recognize both objects and scenery. In addition, the k-means algorithm is employed to find natural scene categories from the extracted visual features. To evaluate their approach, the performance of a system built on the extracted visual features is compared with a system built with low-level features, as well as a hybrid of the two. The aspects considered include accuracy, decision support, and diversity. This is evaluated using metrics such as MAE, precision, and recall, in addition to serendipity measuring techniques. Their results show that the deep learning-based approach outperformed low-level features on all metrics as well as diversity. This indicates that using CNNs for feature extraction in CBF could perhaps constitute as an even more suitable approach than low-level features.

Sulthana et al. [118] demonstrate automation of image processing and analysis for recommendations through the use of a VGG16 CNN model that has been pre-trained on ImageNet. Their approach is, however, not to classify the images and generate recommendations based on image classifications. Instead, they extract feature vectors by disconnecting the base of the model from the classification layers, having the base CNN model analyze similarity relationships between images and using dimensionality reduction on the principle variables from the identified similar images. In order to optimize the performance of recommendations, different dimensionality reduction techniques are evaluated. The proposed methodology obtained high quality recommendations without having to treat the CNN model as a "black box" by reducing the feature-maps. Capturing the notion of similarity, their approach proved to be applicable to both music and images.

The video recommender system proposed by Deldjoo, Constantin, Eghbal-Zadeh, Ionescu, Schedl and Cremonesi [34] replaces manually generated metadata with automatically extracted content descriptions. The content descriptions are extracted from audio and visual channels of a video. Used audio features include block-level and i-vector features, while the visual features include both aesthetic visual features and deep learning features. Genre and tag features are used as baselines. The automatically extracted content descriptors show improvement over traditional metadata in both quality and richness. The authors propose a rank aggregation strategy based on Borda count. The rank aggregation strategy outperforms results from traditional Borda count in fusing recommendations from heterogeneous sources. By utilizing movie trailers as input instead of full movies, the recommendation system achieves better versatility and effectiveness. Their proposed recommendation system is comprehensively evaluated through both a system-centric offline evaluation and a user-centric online experiment. The results indicate that multimedia features can serve as a good alternative to metadata, when it comes to both accuracy and beyond-accuracy measures.

The research gap in combining video classification, search, and personalized recommendation into one unified learning framework is addressed by Lee et al. [67]. Their proposed model is a deep network that utilizes audio-visual content of videos and outputs embedding that aids pair-wise video similarity. Visual features are extracted from one frame per second of a video, using a CNN pre-trained on 100 million labeled images. Frame-level features are aggregated into video-level through average pooling. Audio features are also extracted with deep learning. The extracted features are then fed into an embedding network in order to predict the collaborative signals between videos. Results from their experiments indicate improvement over state-of-the-art on all baselines. The approach is verified to generalize well with various problems, such as video classification and recommendation. Scalability issues are also addressed, and the proposed model is evaluated on large datasets.

Elahi et al. [39] demonstrate use of the off-the-shelf SaaS image recognition tool Rekognition to extract visual features for video recommendation. Utilizing key frames from movie trailers as input, the tool, which is based on deep learning techniques, produces tags or labels of different types of aspects of the key frames, i.e. celebrity name, object label, and face attributes. These visual features were used to train a pure CBF recommender model, and compared with models based on manual tags as well as automatic low-level features. The presented results show that the model trained on automatic visual features from the deep-learning tool Rekognition outperform both manual tags and low-level visual features in predicting user preferences.

2.3 Closed captions for video recommendation

Recent developments in speech recognition has enabled high quality automatic captioning of multimedia content and is in use on streaming platforms such as YouTube [1, 49, 95]. While this may present a novel source of automatic content features, the research opportunity remains highly unexplored. At the same tame, existing textual features used in movie recommender systems are mostly dependent on manually created metadata (i.e. genre, tag).

In music recommender systems, lyrics have been utilizied as a content feature with several approaches. McFee and Lanckriet [80] use lyrics as a part of their automatic playlist generator by connecting songs of the same topic derived from latent Dirichlet allocation (LDA). Similarly, Lim, Lanckriet and McFee [72] learn a similarity function on song-level determined by topic models from bag-of-word representations of song lyrics. Çano and Morisio [20], Laurier, Grivolla and Herrera [66], and Mihalcea and Strapparava [82] explore the use of lyrics in classifying songs by sentiment and mood. In Gossi and Gunes [48], recommendations based on lyrics were found to provide higher performance than collaborative filtering for predicting song categories of musical genres and moods. Similarly to song lyrics, it could be viable to explore how subtitles can be used to extract mood and sentiment of a movie, which would be relevant to context-aware movie recommendation systems [114].

Bocanegra et al. [13] use a semantic content-based recommendation technique which embeds subtitles to enrich YouTube health videos. The system recognizes medical terms in the closed captions and recommends relevant health educational websites to the consumer. A total of 253 recommended links from 53 videos were evaluated by the 253 health professionals who participated. While this approach is context-specific to health videos, it demonstrates an approach that is enabled by subtitle-based recommendations.

2.4 Online evaluation of recommender systems

Algorithmic accuracy and precision has traditionally been the main method of evaluation in the field of recommender systems [61]. However, the sole purpose of recommender systems is to provide users with personalized content that assists them in discovering relevant content [61]. The assumption that high algorithmic performance results in better systems for the user has, in fact, been found to not necessarily always be correct. The most accurate algorithm in the user study by McNee et al. [81] was found to provide the least satisfying results by users, while the most accurate model in Torres et al. [120] provided the least helpful recommendations according to the study participants. Despite the results of these studies and a general consensus that there should be a shift toward user-centric studies that go beyond offline evaluation in recommender systems research [63], few papers test the effect of new recommender system solutions on user satisfaction.

Since a user's satisfaction with a recommender system cannot be measured on its ability to provide accurate recommendations alone, measures that go beyond accuracy are needed for more robust evaluations [61, 62, 81]. Knijnenburg et al. [62] propose a user-centric evaluation framework which identifies six interrelated conceptual components that can be used to explain and predict user behavior in a recommender system (Figure 2.3). The Objective System Aspects (OSAs) constitute the aspects that are up for evaluation. Subjective System Aspects (SSAs) include the perceptions users have of the OSAs, and are measured during or after the interaction with the system through questionnaires. The User Experience factors (EXPs) are the evaluations of the recommender system's qualities from the perspective of the user. This aspect is also measured with questionnaires. The users' interaction with the system (INT) are objective measures of user behavior, i.e. logging of clicks, time spent on certain tasks, etc. Personal Characteristics (PC) and Situational Characteristics (SC) are also factors that may influence the outcome of the evaluation.

Psychological factors play an important role in how people use a system and what they are looking for in it. Personality has a large effect on human decision-making, which is why Tkalcic and Chen [119] argue for its utility for yielding a better picture of a recommender system when assessed in a user-centric evaluation. Cold start situations with new users may also be addressed with the use of personality information. The Five Factor Model of personality, also referred to as the Big Five model, is a comprehensive and widely used personality model in recommender systems. The model identifies five dimensions of personality, namely openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. Table 2.1 displays examples of adjectives related to the Five Factor Model. Several studies show the strong relation between personality and user preferences. Not only does this include categorical and theme preferences [22, 100, 101], but also the composition of a recommendation list [97]. Rawlings and Ciancarelli [97] found users with high openness to prefer diverse styles of music, while the level of extraversion was linked with a user's preferences to popular music.

As the user's perception of a recommender system not only includes the recommendations, but also the design and usability of the interface [62, 96], these aspects must be emphasized when creating an evaluation framework for recommender systems. One example of how layout may affect results is demonstrated in Bollen et al. [14], where users are shown to pay more

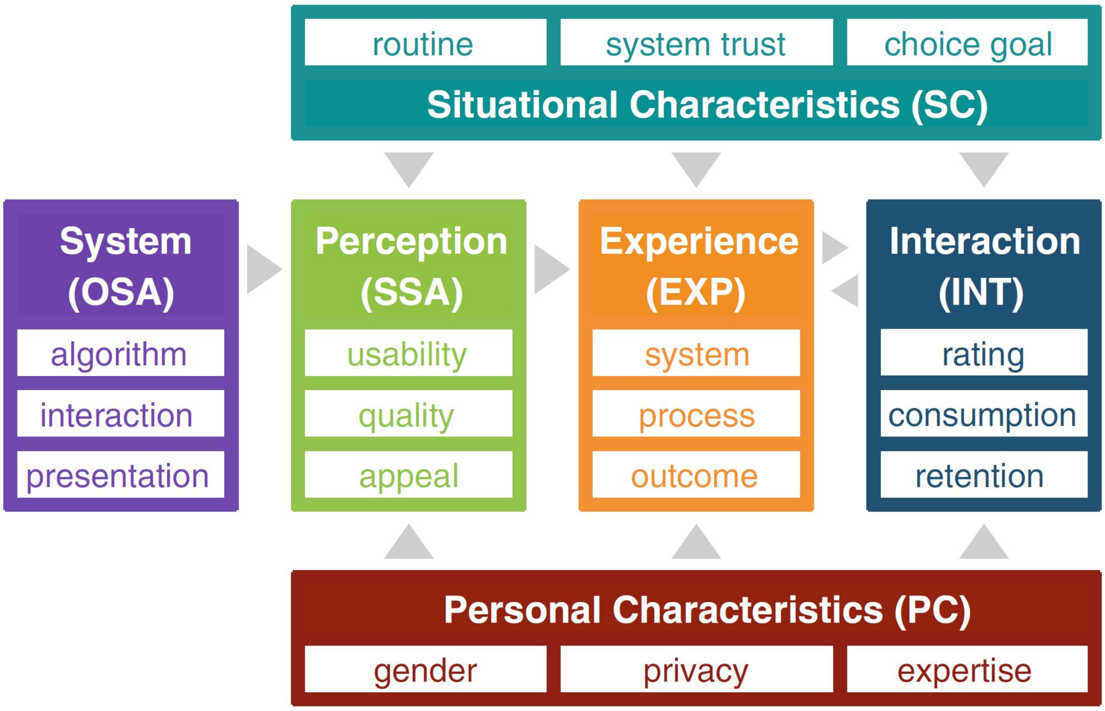


Figure 2.3: User-centric evaluation framework proposed by Knijnenburg et al. [62].

Factor	Adjectives
Extraversion	Active, assertive, energetic, enthusiastic, outgoing, talkative
Agreeableness	Appreciative, forgiving, generous, kind, sympathetic, trusting
Conscientiousness	Efficient, organized, planful, reliable, responsible, thorough
Neuroticism	Anxious, self-pitying, tense, touchy, unstable, worrying
Openness	Artistic, curious, imaginative, insightful, original, wide interest

Table 2.1: Examples of adjectives related to the Five Factor Model [79].

attention to the first few of the items in a vertical list, while paying less attention to items that are lower on the list. While decay is less in a grid layout, users perceive items in the top left of a grid to be most relevant [59]. In systems where users have to toggle between two pages to access different lists of items, few of the items on the second page are chosen by users [24].

Kortum and Oswald [64] employ the System Usability Scale and the Mini-IPIP scale to investigate the impact of Big-five personality traits on the perceived usability of digital products. The study considers 20 different systems which were assessed by 268 users. Indeed, certain personality traits did correlate with the provided usability rating of products. Openness and Agreeableness stood out as the personality traits most tightly linked with perceived usability of a system.

Deldjoo, Schedl and Elahi [29] demonstrate a content-centric web based framework for movie recommendations powered by a content-based model that exploits audio and visual features in addition to metadata. It shares many common functionalities of commercial recommender systems, such as Netflix, letting users rate, search, and browse movies. A new user in the system will be asked to provide demographics and background information and complete a five factor personality assessment. Furthermore, in order to elicit preferences, the user will be invited to select a favorite genre and select four movies and rate selected trailers. In their demo, the implemented model utilizes user preferences and content based features, but the system is highly extendable to other scenarios. The framework can easily be set up to facilitate execution of empirical studies, and by embedding questionnaires for a variety of user characteristics such as demographics and personality, it can also serve as a platform for testing personalized recommendation algorithms. The source code of the framework is freely available online to use [29].

Ekstrand et al. [38] conduct a real-user study on the MovieLens platform to discover perceived differences between three different recommender models. The study is composed using within-subject design where users are asked to compare two lists of recommendations produced by two of the models. In comparing the lists, users answer a 23-item questionnaire in order to assess the perceived differences between the lists on different quality metrics, including accuracy, diversity, satisfaction, and novelty. The study showed that less popular movies were more often perceived as novel. In addition, diversity was positively correlated with satisfaction, while novelty affected satisfaction negatively.

2.5 Summary of Previous Works and Key Differences

This chapter has provided an overview of the existing literature related to the research problems of this thesis. Using visual features to alleviate the cold-start problem has been explored and evaluated in several research papers [29, 33, 39, 41, 84]. Low-level visual features have demonstrated good results, but the results of visual features extracted with deep learning indicate that this approach may have advantages in terms of recommendation quality [42]. Key frames from movie trailers as input to visual feature extraction has been shown to serve well as visual representations of movies [32]. The textual semantic features traditionally utilized in movie recommender systems are heavily reliant on manual labor. Subtitles, which can be generated automatically, represent a novel data source for recommendation that is not explored in the literature. The standards and design of the recommender system interfaces and evaluation frameworks described in this chapter serve as a foundation for this thesis' proposed evaluation framework.

While existing approaches use visual features in pure CBF systems to evaluate the recommendation capabilities, the recommendation technique proposed in this thesis combines user interactions and visual features to generate the user profiles in a hybrid recommender system. To the author's knowledge, a comprehensive evaluation of a hybrid recommender system utilizing deep learning-based visual features, in both offline and online setups, has not really been explored before. Additionally, this thesis includes a large dataset of 12,875 movies, which is a higher number than other previous works that utilize deep learning-based visual features in movie recommendation. Moreover, subtitles, as another novel source of data for movie recommendation, is used as a baseline in the evaluation of the proposed hybrid recommendation technique together with more traditional features (i.e. tag and genre). The proposed evaluation framework for online evaluation is mainly serving as a tool for the real-user study. In addition to this, the effect of user personality on perception of users (e.g., on usability) is addressed in the context of a hybrid recommender system based on visual features.

Chapter 3

Methodology

This chapter details the techniques and procedures that were used to address the defined research questions for the thesis. Section 3.1 describes the the feature extraction methods utilized to form the datasets that are used for training the recommendation models. The section includes a description of how a pre-trained CNN model was utilized to automatically extract visual features from 12,875 movie trailers to form a novel dataset for movie recommendation. In addition, the accumulation of subtitles is detailed. Section 3.2 shows the steps involved in pre-processing and aggregating the extracted item features to form three separate sets of feature vectors. Section 3.3 provides a description of the recommender system approach and the algorithms used in training the prediction models used in the evaluation of my research. The technical details and design of the prototype recommender system interface used in the real-user evaluation is elaborated in Section 3.4. Finally, the conditions and metrics selected to evaluate the approach, as well as an overview of the statistical analysis methods are given in Section 3.5.

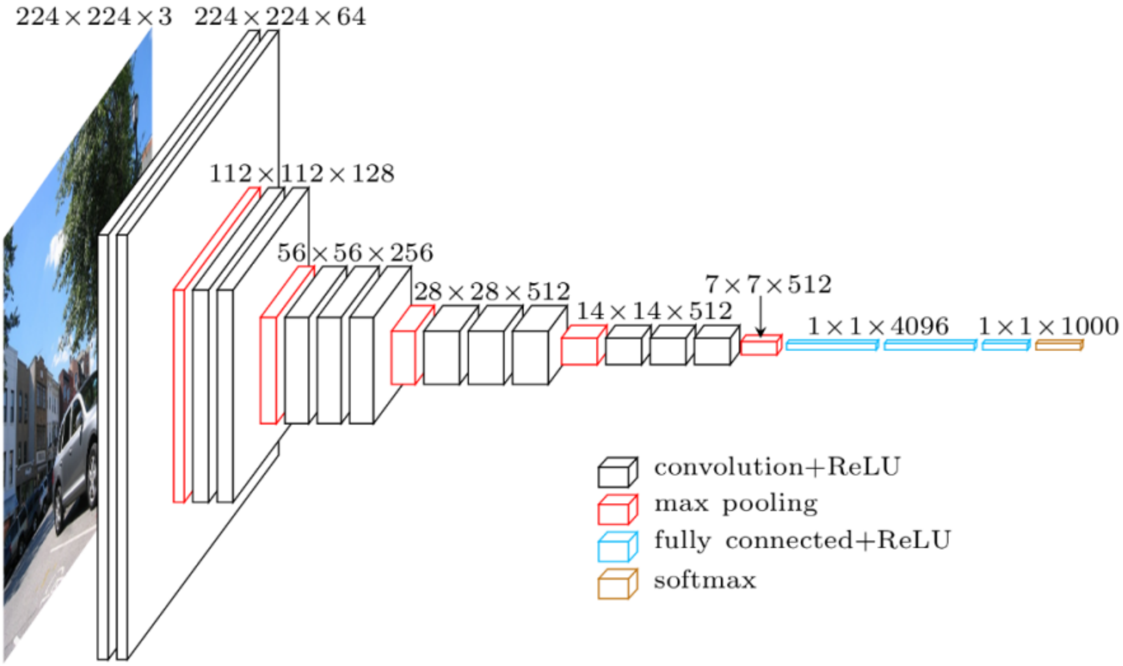


Figure 3.1: Architecture of the utilized VGG-19 convolutional neural network [11, 115]

3.1 Feature Extraction

The feature extraction can be divided into two parts. The first part includes the extraction of visual features from 12,875 movie trailers using convolutional neural nets (CNN). The second part encompasses the collection of movie subtitles.

3.1.1 Visual Feature Extraction

Visual features have been extracted by applying the VGG-19 image classification model [115] to the key frames of every movie trailer in the key-frame dataset. The VGG-19 model is a state-of-the-art deep CNN for image classification which utilizes its 19 weight layers to produce class labels from image input. The model was trained on ImageNet and applied to the key frames to produce class labels of the images, serving as visual features for the purpose of movie recommendation. Having movie trailers reduced to sets of key frames saves a significant amount of computational power needed for the feature extraction.

The dataset of key frames serving as the source from which visual features are extracted, is based on the work of Moghaddam et al. [84] and Elahi et al. [41]. By applying techniques based on color histogram distance, videos were split into building blocks of shots. A shot is denoted as a sequence of successive frames captured with no interruption by the film camera. Within each shot, a frame has been selected as the representative key frame. Since transitions between shots in movies are typically abrupt, they can be identified by looking at the color histogram intersection between each frame. If h_t and h_{t+1} are denoted as histograms of consecutive frames and b is the index of the histogram bin, the intersection can be computed as [84]:

$$\mathbf{s}(h_t, h_{t+1}) = \sum_b min(h_t(b), h_{t+1}(b)) \tag{3.1}$$

One single frame (key frame) is chosen as a representation of each identified shot. The resulting dataset is comprised of a total of 2,446,561 key frames across 12,875 movie trailers. Each movie is identified with an id corresponding to the same movie in the MovieLens dataset.

The VGG-19 image classification model was implemented in Python, using the Keras API, which is built on top of the TensorFlow framework [87]. Pre-trained on more than 1.2 million images from ImageNet [107], the output of the model consists of a label, representing the predicted classes of the input image, as well as a confidence value representing the certainty of the prediction being correct. Figure 3.2 shows an example of key frames and the predictions made by the model. The resulting dataset of labels for 12,875 movies includes 997 unique feature labels in total. Even though only using key frames significantly reduces the amount of frames to be analyzed, image recognition with CNNs is still a computationally heavy task. The feature extraction process was carried out in Google Colaboratory[1], which gives free access to computing resources, including GPUs. The hardware included Nvidia Tesla T4 16GB 2560 GDDR6 GPU, Intel Xeon 2.20GHz CPU, and 13GB RAM.

3.1.2 Subtitle Collection

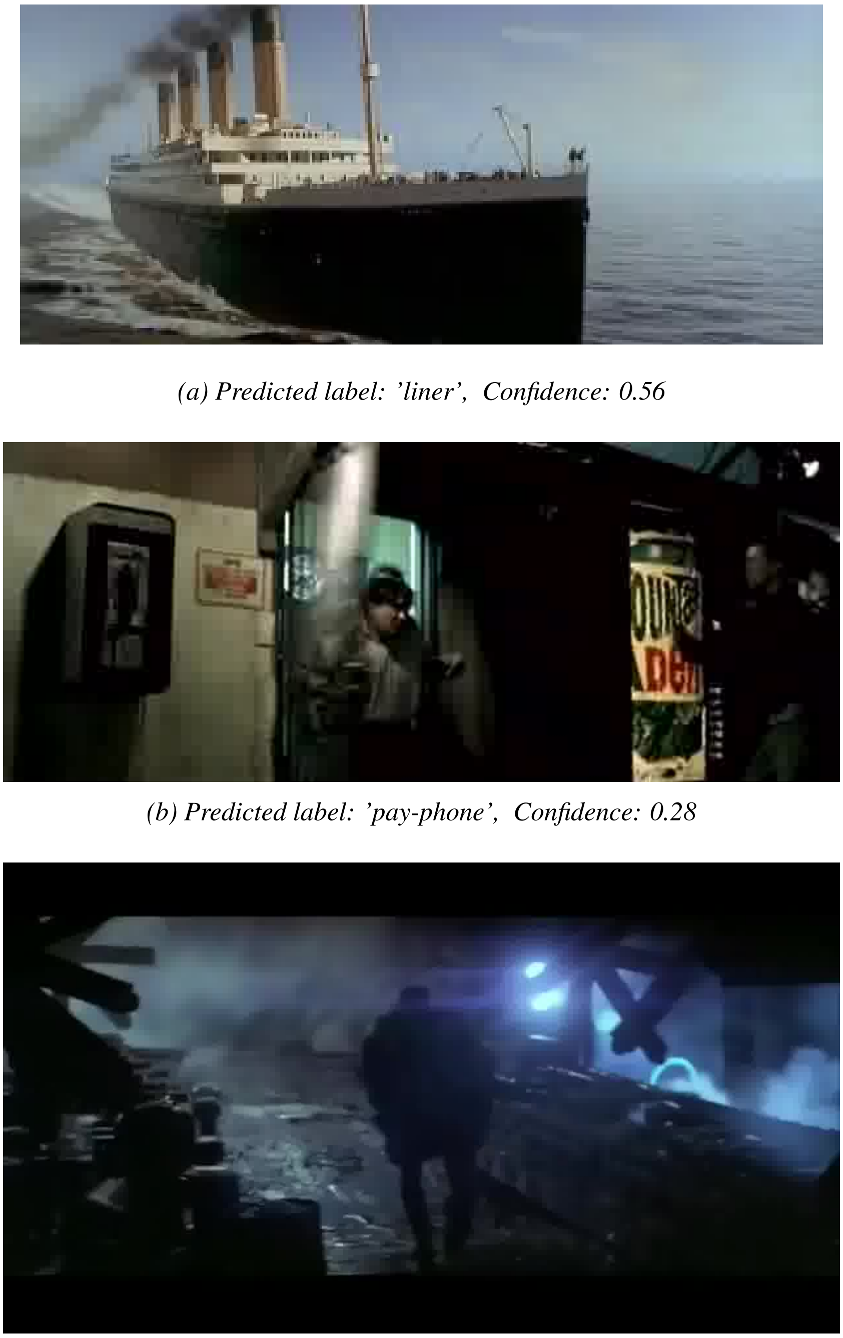
As a fully automatic feature for video recommendation, subtitles should be extracted from automatically generated closed captions. Even though speech recognition for movie feature extraction is beyond the scope of this thesis, subtitles were included as a comparison baseline to demonstrate the potential capabilities of this type of automatic feature. Accordingly, English subtitles for 3,405 full movies were collected using a public API [88] [2].

3.2 Feature Aggregation

Features of the different datasets were aggregated to form Vector Space Models, which were used as input for the recommender models. In vector space models, keywords are represented by a vector in an n-dimensional space in which each of the dimensions corresponds to a term in the global vocabulary of a collection of documents. The documents are represented as vectors of term weights, where weight is an indicator of the degree to which the document and the term are associated. A vector space model can formally be described with $D = \{d_1, d_2, , ..., d_N\}$, denoting a set of documents, and the set of words in the overall corpus, i.e. the dictionary, denoted as $T = \{t_1, t_2, , ..., t_N\}$, with $d_j = \langle w_{1j}, w_{2j}, ..., w_{nj} \rangle$ denoting the representation of

[1] colab.research.google.com/

[2] opensubtitles.org



(a) Predicted label: 'liner', Confidence: 0.56

(b) Predicted label: 'pay-phone', Confidence: 0.28

(c) Predicted label: 'spotlight', Confidence: 0.15

Figure 3.2: CNN label predictions of example key frames from three different movie trailers: (a) Titanic, (b) Fight Club, and (c) Blade Runner

each document d_j in the n-dimensional vector space where w_{kj} represents the weight of the term t_k in document d_j [27].

In the vector space models used in this work, a movie is considered a document. The terms refer to feature labels in the visual feature dataset, words in the subtitle dataset, genre in the MovieLens genre dataset, and tag in the MovieLens tag datset. Since the genre and tags are unique, the weights for these are binary, while weights for visual features and subtitles were aggregated, as labels or words may occur several times for each movie. Visual features were aggregated using two different methods, producing two separate vector space models, *DeepCineProp-f* and *DeepCineProp-c*.

DeepCineProp-f. Visual features were weighted using *Term Frequency–Inverse Document Frequency (TF-IDF)* [58]. TF-IDF can recognize the importance of each word in a document in the context of a corpus of documents, and is one of the most widely used weighting schemes in CBF research [9]. If a term has low occurrence across the corpus while having high frequency in one (or few) documents, it likely plays a key role in that specific document. The TF-IDF formula can be defined as:

$$\text{TF-IDF}(t_k, d_j) = \text{TF}_{t_k,d_j} \cdot log(\frac{N}{n_k}) \tag{3.2}$$

where TF_{t_k,d_j} refers to the number of occurencies of term t_k in the document d_j, n_k is the number of documents that contain t_k, and N is the total number of documents. In this case, a movie is considered a document, and the labels of the movie are considered to be terms of that document. Furthermore, the collection of all movies and their respective labels correspond to the corpus of documents.

DeepCineProp-c. Important elements in a movie can be assumed to be emphasized visually, and thereby more likely to be predicted with a higher confidence, computed by the image classification model. Based on this assumption, visual features were weighted according to the mean confidence value of each label occurring in a movie to form the DeepCineProp-c dataset. Figure 3.2 displays examples of labelled key frames and the confidence which was used as weight in DeepCineProp-c.

CineSub. Subtitle features were parsed and pre-possessed, resulting in a dataset of English subtitles from 3,405 different movies. Since the main interest of using subtitles as a source of content features for a recommender model are the actual words said in the movies, subtitle specific data, such as timestamps and URLs were removed from the documents. Furthermore, to transform the raw subtitle data into cleaner information and reduce the size of the dataset, a number of pre-processing techniques were applied [122]. First, the content of each document was tokenized, meaning that the text was split into individual words. As stop words are generally seen as less important in text analysis, these were removed. Lemmatization was used to transform words into their inflected forms, i.e. the dictionary form of the meaning of the word. This step significantly reduces the dimensionality of the dataset. To reduce the dimensionality of the dataset further, part-of-speech filtering was applied, removing from the documents all words that are not nouns. This final step makes the resulting vector space model include terms similar to those of the DeepCineProp vector space models, which contain terms referring to objects. While methods for further refining should be assessed to reduce computational load for training recommender models based on subtitle features, the steps taken are sufficient within the scope of this thesis. The resulting vector space model based on subtitle features, CineProp, includes 62,664 unique features. As with DeepCineProp-f, the CineSub features were weighted using the popular TF-IDF method.

3.3 Recommendation Algorithm

The recommender model used in the experiments extends the Matrix Factorization model and is able to learn different types of user and item representations. Hence, the model is capable of taking advantage of heterogeneous data, including different types of side information (e.g. visual features, subtitles, genre of movies, tags of users). The implementation of the hybrid recommender model has been done using a popular open-source library, LightFM [65]. This library offers a state-of-the-art hybrid latent representation recommender model which can be implemented with one of several available optimization algorithms.

To formally describe the model, let I represent the set of items, U represent the set of users, F^I represent the set of item features, and F^u represent the set of user features. Users have interactions with items that are either favourable (positive interactions) or unfavourable (negative interactions). The union of both positive S^+ and negative S^- interactions form the set of every user-item interaction pair $(u,i) \in U \times I$. An item i is represented by a set of features $f_i \subset F^I$. The same is the case for a user u whose features are represented by $f_u \subset F^u$. For each feature f, the model is represented in terms of d-dimensional item and user feature embeddings e^I_f and e^U_f. A scalar bias term b^I_f for item and b^U_f for user features is also included in describing a feature. The sum of the latent vectors of its features gives the latent representation of item i:

$$q_i = \sum_{j \in f_i} e^I_j \tag{3.3}$$

The same is the case for user u:

$$q_u = \sum_{j \in f_u} e^U_j \tag{3.4}$$

The sum of biases of its features gives the scalar bias term of item i:

$$b_i = \sum_{j \in f_i} b^I_j \tag{3.5}$$

The same is the case for user u:

$$b_u = \sum_{j \in f_u} b_j^U \tag{3.6}$$

The model then makes predictions for user u and item i by taking the dot product of item and user representations, adjusted by item and user biases:

$$\hat{r}_{u,i} = f(q_u \cdot p_i + b_u + b_i) \tag{3.7}$$

where $f\cdot$ is given by the sigmoid function:

$$f(x) = \frac{1}{1+\exp(-x)}. \tag{3.8}$$

Different methods of optimization may result in substantially different outcomes in recommendation. This may be influenced by e.g. the nature of the features available to the model. Since the vector space models of content features used in the experiments are dissimilar in terms of dimensionality, sparsity, and type, three different optimization methods with different loss functions have been considered: *Weighted Approximate-Rank Pairwise (WARP)* [126]; *Bayesian Personalized Ranking (BPR)* [99]; and *Logistic*.

The WARP loss function is defined as [55, 126]:

$$Err_{\text{WARP}}(\mathrm{x}_i, y_i) = L[rank(f(y_i|\mathrm{x}_i))] \tag{3.9}$$

where the function $rank(f(y_i|\mathrm{x}_i))$ measures the number of negative labelled instances that are "wrongly" given a higher rank than this positive example x_i :

$$rank(f(y_i|\mathrm{x}_i)) = \sum_{(\mathrm{x}',y') \in C_u^-} \mathbb{I}\left[f(y'|\mathrm{x}') \geq f(y|\mathrm{x}_i)\right] \tag{3.10}$$

where $\mathbb{I}(\mathrm{x})$ is the indicator function, and $L(\cdot)$ transforms this rank into a loss:

$$L(r) = \sum_{j=1}^{r} \tau_j, \text{with } \tau_1 \geq \tau_2 \geq \cdots \geq 0. \tag{3.11}$$

This class of functions allows one to define different choices of $L(\cdot)$ with different minimizers. Minimizing L with $\tau_1 = 1$ and $\tau_{i>1} = 0$, the precision at 1 is optimized, $\tau_j = \frac{1}{Y-1}$ would optimize the mean rank, while for $\tau_{i \leq k} = 1$ and $\tau_{i>k} = 0$ the precision at k is optimized. For $\tau_i = 1/i$, a smooth weighing is given, where the top position is given more weight, with rapidly decreasing weights for lower positions. This is useful when optimizing Precision@K for a range of different values at K is desirable.

BPR [99] is one of the state-of-the-art algorithms exploiting homogeneous implicit feedbacks. It assumes that a user prefers a consumed item to an unconsumed item, denoted as

$(u,i) \succ (u,j)$ or $\hat{r}_{uij} > 0$. Mathematically, BPR solves the following minimization problem [92, 99]:

$$\min_{\Theta} \sum_{(u,i,j):(u,i)\succ(u,j)} f_{uij}(\Theta) + \mathcal{R}_{uij}(\Theta) \tag{3.12}$$

where the loss function $f_{uij}(\Theta) = -\ln \sigma(\hat{r}_{uij})$ is designed to encourage pairwise competition with $\sigma(\mathrm{x}) = 1/(1+\exp(-\mathrm{x}))$ and $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$. Note that $\mathcal{R}_{uij}(\Theta) = \frac{\propto}{2}\,\|U_{u\cdot}\|^2 + \frac{\propto}{2}(\|V_{i\cdot}\|^2 + \|V_{j\cdot}\|^2) + \frac{\propto}{2}(\|B_i\|^2 + \|B_j\|^2)$ is the regularization term used to prevent overfitting, and $\hat{r}_{ui} = \langle U_{u\cdot}, V_{i\cdot}\rangle + b_i$ is the prediction rule based on user u's latent feature vector $U_{u\cdot} \in \mathbb{R}^{1\times d}$, item i's latent feature vector $V_{i\cdot} \in \mathbb{R}^{1\times d}$ and item bias $B_i \in \mathbb{R}$.

Even though logistic regression is not widely spread in the literature of recommender systems, it is common in the industry, perhaps due to its efficiency and simplicity [6]. The logistic loss function can be denoted as [93]:

$$\underset{U,M,C}{min} \sum_{i}^{n} \sum_{j}^{m} [w_{ij}(p_{ij} - \langle U_{i*} M_{j*}\rangle)^2 + \frac{\lambda}{n}||U_{i*}||^2 + \frac{\lambda}{m}||M_{i*}||^2] \tag{3.13}$$

where w_{ij} marks the confidence value of user-item interactions.

3.4 Prototype

I have built a demo application for evaluating movie recommender systems, called *SAMVISE*. The application is completely web-based and is designed for running on a wide range of devices, such as smartphones, personal computers, and tablets. While serving as a viable hybrid movie recommender system, the main contribution lies in its utility as a modern framework for evaluating different recommendation algorithms in the movie domain. Although the framework is developed from scratch for the purpose of this thesis, the proposed framework of Deldjoo, Schedl and Elahi [29] and the study by Ekstrand et al. [38] served as inspiration. In the following, a description of the interface in terms of implementation and design is given, succeeded by an account of the use of the system.

3.4.1 Technical details

The framework was implemented as a completely web-based application using the popular JavaScript libraries React and NodeJS, with MongoDB Atlas as the database system. The recommender models for the user study were implemented as an external Python Flask API. The system's architecture is represented in Figure 3.5, and easily enables querying of other movie recommender APIs for evaluation purposes. Considering that nearly half of the traffic from the world's active internet users comes from mobile devices [25, 116], the framework's layout was designed to work well for these devices' typical screen resolutions.

3.4.2 Interface

The interface and content are created minimalistic and simple in order to make the user experience smooth and prevent confusion and distractions from the tasks at hand. Additionally, technical terms are kept to a minimum to keep users from feeling that they are not knowledgeable enough to participate [111]. Other functions implemented in order to enhance the user experience include a progress bar indicating how far the participant has proceeded in the study, interactive elements, as well as designing for accessibility [18].

In the user interface of the framework, as shown in Figures 3.3 and 3.4, questionnaires (i.e. demographics, Ten-Item Personality Inventory, recommender evaluation) are presented as single-item with either likert-scale, drop-down, or numerical input. In the rating elicitation, movies are displayed in a grid, with options for users to filter by decade and sort by either popularity or rating in order to find movies they know. The two recommendation lists presented to users for evaluation are displayed in separate rows. Both the grid view for browsing videos and the row structure for displaying separate lists of movies are similar to what is standard in many popular video recommender interfaces these days (e.g. YouTube and Netflix) [5, 45].

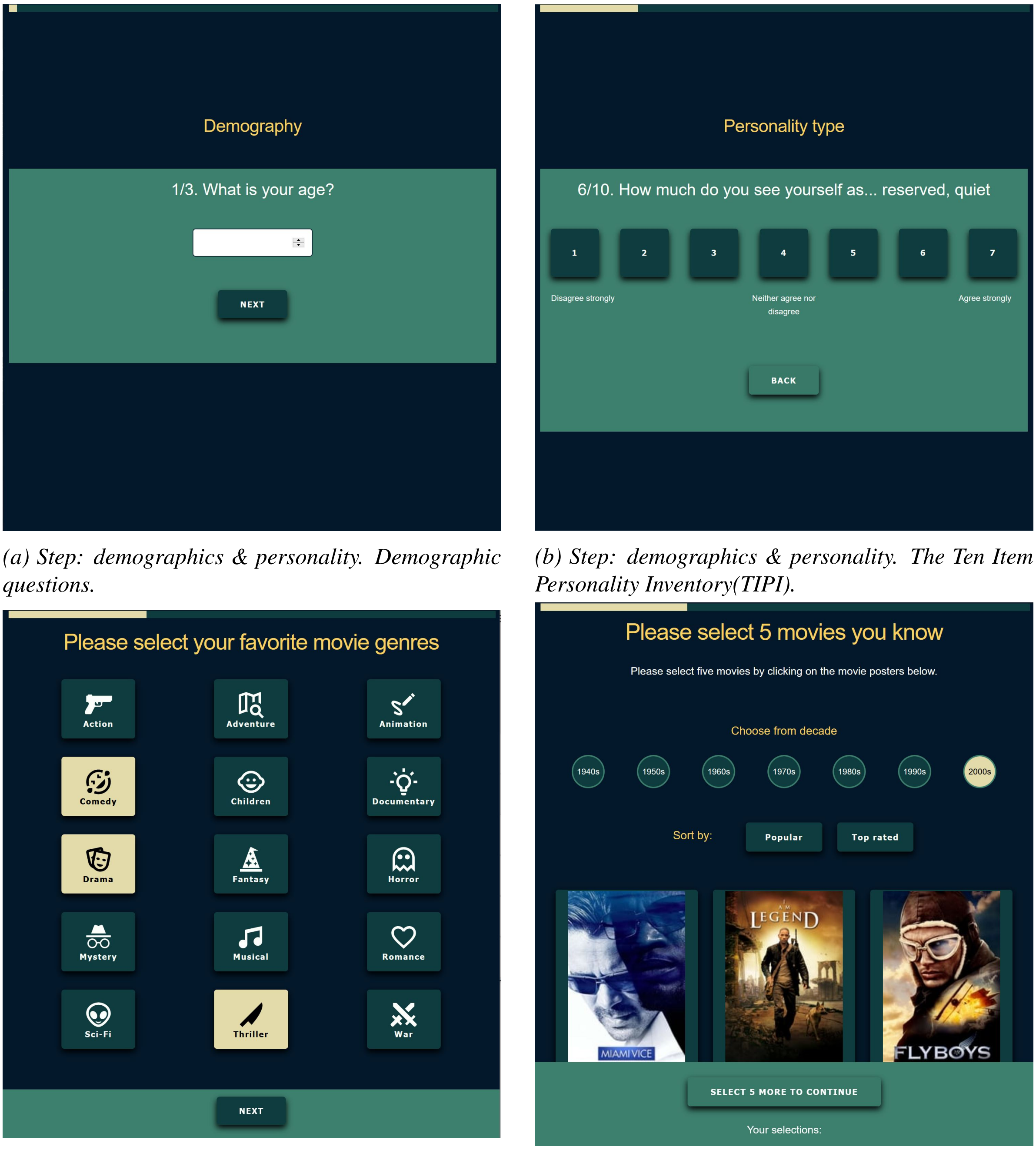
3.4.3 Steps

The user study is divided into five steps, namely *instruction*, *demographics & personality*, *movie selection & rating*, *recommender evaluation*, and *usability evaluation*. The different steps are described in the following:

Instruction. In this step, participants are given information about what tasks they will be given. For transparency, participants are also informed of what types of data are collected and how the data are handled. Based on the information, participants may proceed to enter the study.

Demographics & personality. Participants are asked general demographic questions, as well as the Ten Item Personality Inventory (TIPI). Within the limitations of this thesis, demographic and personality data are gathered for data analysis purposes only. However, these data could be utilized as user features for recommendation purposes [12, 86].

Rating elicitation. In this step, participants are asked to select at least five movies they know, and subsequently rate the movies on a scale from 1-5 [26]. Participants can find movies they have enough knowledge about to rate by using the functions of filtering movies by decade and sorting after popularity or rating. The participants have the option to watch the trailer and read information (e.g. plot and credits) about the movies (if needed) before providing a rating. The design of this step is displayed in Figures 3.3c, 3.3d, and 3.4a. The models in the recommender component each produces one list of top-N recommendations, based on



Demography

1/3. What is your age?

Personality type

6/10. How much do you see yourself as... reserved, quiet

Please select your favorite movie genres

Please select 5 movies you know

Please select five movies by clicking on the movie posters below.

Your selections:

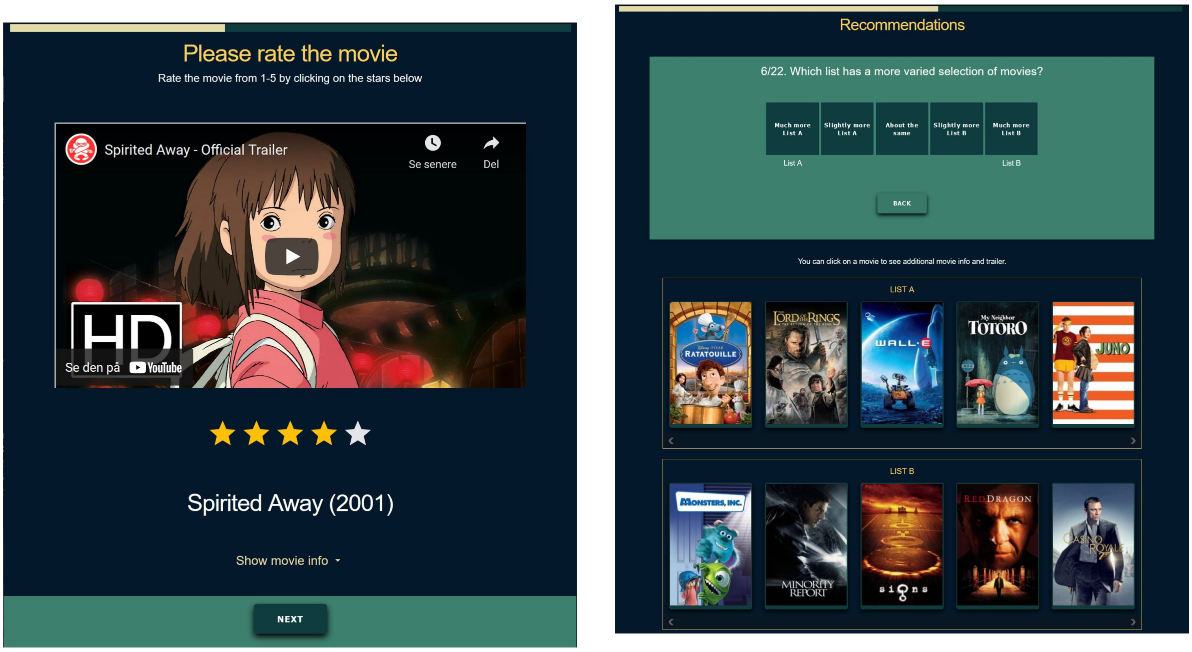
(a) Step: demographics & personality. Demographic questions.

(b) Step: demographics & personality. The Ten Item Personality Inventory(TIPI).

(c) Step: rating elicitation. Genre selection to find movies the user is interested in.

(d) Step: rating elicitation. Selection of the movies to rate.

Figure 3.3: Screenshots from the interface of the movie recommendation evaluation framework


Please rate the movie

Rate the movie from 1-5 by clicking on the stars below

Spirited Away (2001)

Recommendations

6/22. Which list has a more varied selection of movies?

You can click on a movie to see additional movie info and trailer.

(a) Step: rating elicitation. The user provides their rating of selected movies.

(b) Step: Recommender model evaluation. The user is asked questions comparing the two presented lists of recommendation.

Figure 3.4: Screenshots of the interface in different steps and procedures of the movie recommendation evaluation framework.

ratings provided by the active participant. The configurations of the recommender model in the recommender component of the real user study were similar to the settings described in Section 3.5.1. The models included were based on DeepCineProps-f, CineSub, and tags, where CineSub and tags served as baseline models.

Recommender model evaluation. A/B testing, more specifically a within-subjects study design, is used in the recommender model evaluation [50, 62]. In this study design, each participant is tasked with comparing two randomly picked recommendation lists out of the three lists generated by the recommender component (Figure 3.5), i.e. recommendations based on DeepCineProps-f, CineSub, and tags. The randomization contributes to a likelihood of even distribution recommendation models presented to users. In order to avoid possible biases [111], each selected recommender list is randomly assigned one of the titles "List A" or "List B" every time they are loaded. In addition, the order of which they are presented in the interface is randomized. The choices made in this step in regards to questions asked and comparison of different recommendation lists are based mainly upon the study in Ekstrand et al. [38] and Knijnenburg et al. [62] which was executed on the MovieLens platform. The section includes a total of 21 questions to measure the user's perception of the recommendation lists. These questions are related to five different aspects of the lists further explained in Section 3.5.2, i.e. accuracy, satisfaction, perceived personalization, novelty, and diversity [38].

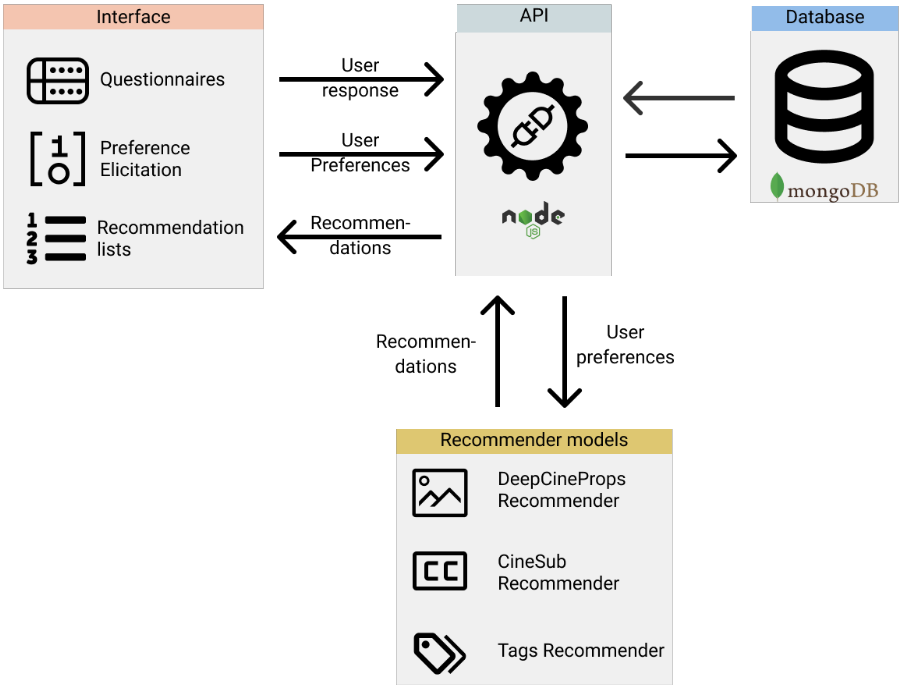


Figure 3.5: The overall architecture of the demo for the evaluation framework.

Usability evaluation. In the final step of the user study, participants are tasked with evaluating the usability of the system by responding to the System Usability Scale(SUS) questionnaire [15]. The SUS is a simple and reliable 10-item questionnaire for overall subjective assessment of a system's usability, further described in Section 3.5.2.

3.5 Experiment design

The research design for evaluating the research approach in regards to the research questions is detailed in this section. This includes a description of the methodology of the offline experiments, in addition to an elaboration of the design of the real-user study.

3.5.1 Offline Evaluation

An initial exploratory analysis was performed on the DeepCineProp13K dataset, using the dimentionality reduction, clustering, and topic modeling. For dimentionality reduction, the techniques Principal Component Analysis (PCA)[127] and t-distributed stochastic neighbor embedding (T-SNE)[121] were utilized. Clustering was performed with K-means and topic modeling with Latent Dirichlet Allocation (LDA)[10]. Included in the exploratory analysis were 2944 items for DeepCineProp. Dataset sizes in the data exploration were reduced for performance purposes as well as due to lack of metadata for some movies (i.e. release date, popularity, and genre).

The proposed recommendation techniques were evaluated based on (automatic) visual features considering different optimization methods, i.e., WARP, BPR, and logistic loss functions utilizing both item features and user interactions. Each model was trained on one of the two types of automatic features (i.e., item embeddings), namely *DeepCineProp-f*, *DeepCineProp-c*. Recommendation based on *CineSub*, *tags*, or *genre* have been considered as baselines. While subtitles can be automatically extracted, both genre and tags require human-annotation. In addition to item features, MovieLens10M dataset [51] has been utilized. In order to simulate the cold-start scenario, I have randomly sampled the dataset. The final result contained 4,368,481 ratings for 3,405 items provided by 69,877 users.

The train and test sets have been built by following a hold-out methodology, i.e., randomly splitting the dataset into 80% (train) and 20% (test) disjoint subsets. The proposed recommendation models have been trained using the train set and evaluated using the test set. Hyperparameter tuning has been performed using a random search to fit LightFM models with random hyperparameter values and evaluating the model performance on the validation set. Based on the hyperparameter tuning result, models were trained over 25 epochs with AdaGrad [37] as learning rate schedule and a learning rate of 0.06.

In the offline evaluation, the measures of performance utilized to evaluate the system include AUC, Recall@K, Precision@K, and Reciprocal Rank:

Area Under the ROC Curve (AUC) measures the ability of a recommender system to distinguish items liked by a user (relevant items) from all the other items (irrelevant items) [76]. [73]. ROC stands for the Receiver Operator Characteristic, and illustrates a plot of recall versus fallout at different threshold settings [54]. We can calculate AUC by comparing the probability of relevant items being recommended with the probability that irrelevant items will be recommended. Through n times of independent comparisons, if there are n' times when the relevant item has a higher score than the irrelevant item and n^n times when the relevant and the irrelevant item have the same score, AUC can be defined as [130]:

$$AUC = \frac{n' + 0.5n^n}{n} \tag{3.14}$$

A perfect AUC score of 1 is achieved when all relevant items have higher scores than irrelevant items, whereas an AUC score of 0.5 would be achieved by a randomly ranked list of recommendations. Consequently, the amount by which the AUC score exceeds 0.5 indicates how well the recommendation algorithm performs in identifying relevant items.

Precision at top K recommendations (P@K) measures the proportion of recommended items in the top-k set that are relevant (true positives + true negatives). The labels of items should be binarized in order to make relevance judgements. For instance, if items are labeled on a five-point Likert scale, ratings greater than or equal to 4 should be labeled as relevant. This measure represents the probability of a recommended item being relevant and is computed as follows [110]:

$$P_u@K = \frac{|L_u \cap \hat{L}_u|}{|\hat{L}_u|} \tag{3.15}$$

where the set of relevant items for user u in the test set T is denoted as L_u and the K items in T with the predicted highest ratings for the user u is denoted as $\hat{L}_u$. The overall $P@K$ is found by averaging $P_u@K$ values for every user u in T.

Recall at top K recommendations (R@K) measures the ratio of $top-k$ recommendations (true positives) and all other relevant but not retrieved items (true positives + false negatives). This measure represents the probability of a relevant item being recommended. $R_u@K$ is computed as [110]:

$$R_u@K = \frac{|L_u \cap \hat{L}_u|}{|L_u|} \tag{3.16}$$

where the set of relevant items for user u in the test set T is denoted as L_u and the set containing the K recommended items in T for u is denoted as $\hat{L}_u$. The overall $R@K$ is found by averaging $R_u@K$ values for every user u in T.

Mean Reciprocal Rank (MRR) is a popular metric used to measure the performance of top-k recommendations. We can arrive at the Reciprocal Rank (RR) for a given list of recommendations of a user by measuring the position (rank) of the first relevant recommended item. The MRR is the average RR across every list of recommendations for each individual user [113]. When the goal is to provide users with few but valuable recommendations, MRR is a particularly useful measure. Reciprocal Rank of a ranked list for user u can be expressed as [123]:

$$RR_u = \sum_{j-1}^{N} \frac{Y_{uj}}{R_{uj}} \prod_{k=1}^{N} (1 - Y_{uk}\mathbb{I}(R_{uk} < Ruj)) \tag{3.17}$$

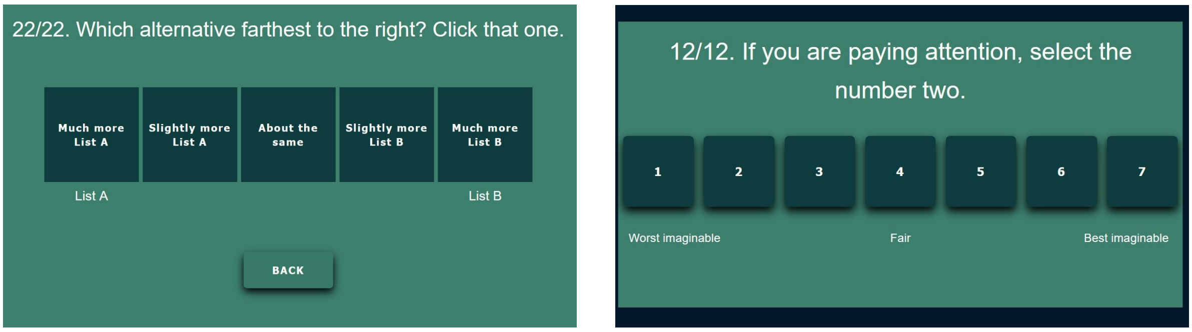
where the number of items is denoted as N, the binary relevance score of item j to user u is denoted as Y_{uj}, i.e. if item j is relevant to user u then $Y_{uj} = 1$, otherwise 0, and $\mathbb{I}(x)$ is an indicator function that is equal to 1 if x is true, 0 otherwise. The rank of item j in the ranked list of items for user u is denoted as R_{uj}.

3.5.2 User Study

The user study involved 166 participants, whereas 48 were voluntary and 118 were engaged through the crowdsourcing platform Prolific[3]. The data were collected over a period of one month, and participants recruited through Prolific were monetarily compensated for their contribution to the study.

Since the user study was implemented in English, pre-screening was applied to the Prolific users, only letting profiles that claim fluent proficiency in English participate. To ensure the data quality and prevent potential "cheaters", two Instructional Manipulation Checks

[3] prolific.co/



22/22. Which alternative farthest to the right? Click that one.

12/12. If you are paying attention, select the number two.

(a) Instructional Manipulation Check 1.

(b) Instructional Manipulation Check 2.

Figure 3.6: Screenshots of the Instructional Manipulation Checks (IMU)[89] used to screen "cheating" participants in the user study.

(IMU)[89] were implemented in order to determine whether the participants were paying attention to the study (Figure 3.6). Users who failed the IMUs were discarded from the final data analysis, resulting in a reduction of 9.6% from 166 to 150 users.

Metrics

The following metrics were assessed in the user-centric evaluation of the proposed recommender technique:

- **Accuracy.** Perceived accuracy measures the level to which a user feels that the recommended movies are "good" and appealing [38].
- **Satisfaction.** Overall satisfaction of the user with the list of recommendations and the degree to which the recommendations fulfill the user's needs or wants [38].
- **Perceived personalization.** The user's perception of the degree to which the recommender model is able to match their personal tastes, interests, and preferences [38, 96]. It is an overall measure of how well a recommender is able to learn and adapt to user preferences and tastes.
- **Novelty.** The degree to which the recommender model provides a user with new and interesting recommendations. Novelty is related to the ability of a recommender to assist users in discovering new items.
- **Diversity.** The level of which a recommender model provides diverse lists of items to choose from. While providing more satisfactory recommendations, a diverse list can make the decision process an easier and more pleasant experience for users [131].

Personality and demographic influences

Previous studies have shown that demographic and personality factors can be linked to user preferences [86]. These factors have been included to discover if they are influential to which

	Factor	Statement: I see myself as...
1.	Extraversion	Extraverted, enthusiastic
2	Agreeableness	Critical, quarrelsome
3.	Conscientiousness	Dependable, self-disciplined
4.	Neuroticism	Anxious, easily upset
5.	Openness	Open to new experiences, complex
6.	Extraversion	Reserved, quiet
7.	Agreeableness	Sympathetic, warm
8.	Conscientiousness	Disorganized, careless
9.	Neuroticism	Calm, emotionally stable
10.	Openness	Conventional, uncreative

Table 3.1: The Ten Item Personality Inventory (TIPI) [47].

recommender model users in this study prefer. The demographic data collected include gender, age, and nationality. The Ten Item Personality Inventory (TIPI) (Table 3.1) is used to assess a participant's personality traits within The Big Five [47]. Personality largely affect human decision-making, and has been found to be strongly correlated with user preferences and decisions in recommender systems [22, 97, 100, 101]. Personality is therefore an important aspect to address in order to yield a more complete picture of a recommender system in a user-centric evaluation.

System Usability

To measure the usability of the system from a user's perspective, the System Usability Scale (SUS) was utilized [15]. The usability evaluation was made optional for the voluntary participants, resulting in a total of 145 SUS responses. Originially developed as a "quick and dirty" usability scale, the SUS has become one of the most widely used post-study subjective assessments of usability [70]. Although the questionnaire is reasonably quick, research has shown that it is not "dirty" [109]. The SUS consists of 10 items, each with a 5-point likert scale. Questions that are odd-numbered have a positive tone, while the tone of even-numbered questions is negative. Lewis and Sauro [70] suggest that we can extract additional information by decomposing the SUS into Usable and Learnable components. However, based on accumulating new evidence, Lewis [69] recommends to treat it as an unidimensional assessment of perceived usability without computing the Usability and Learnability subscales. To better explain the overall user experience, a complementary 11th adjective rating scale question was added as described by Bangor, Kortum and Miller [8].

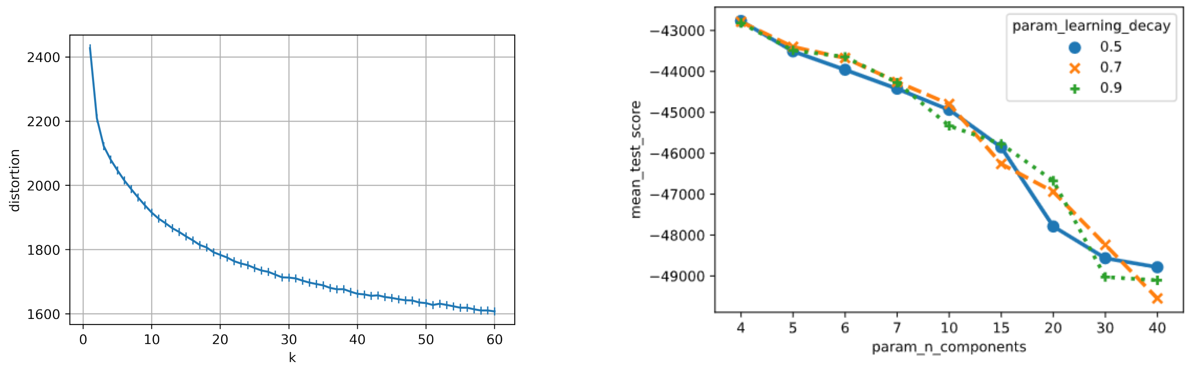
Statistical analysis

In order to obtain more detailed knowledge about how the different models are perceived by users, a statistical analysis calculating the variances between associated metrics was carried out. The student-t test was utilized to find Pearson correlation coefficients and p-values.

Chapter 4

Results

This chapter describes in detail the analysis performed on the accumulated datasets, recommender models, and user study data. It is organized in line with the different experiments that were executed. Section 4.1, **Experiment A: Data exploration**, details the exploratory analysis performed on the DeepCineProp13K dataset. Section 4.2, **Experiment B: Recommendation Quality**, details the offline evaluation of recommendation quality of DeepCineProp-f and DeepCineProp-c compared with different baseline models. Section 4.3, **Experiment C: Comparing Loss Functions**, details the evaluation of how recommender models based on different feature types perform with different loss functions. Section 4.4, **Experiment D: Real-User Study**, details the analysis of data from the real-user study, including both recommendation quality and usability evaluation.



(a) Elbow method DeepCineProp

(b) Grid search for tuning the LDA model

Figure 4.1: Elbow method and grid search performed on the DeepCineProp13K dataset.

4.1 Experiment A: Data Exploration

The initial set of experiments includes an exploratory analysis of the data in order to better understand the datasets. The results of the exploration of DeepCineProp13K are detailed in this section.

In order to visualize the DeepCineProp13K dataset over a 2-dimensional space, the dataset reduction technique t-distributed stochastic neighbor embedding (t-SNE) was performed on the data [121]. T-SNE, which is based on non-convex optimization, is a popular technique to visualize high-dimensional data. Since t-SNE is computationally expensive, and each item in the dataset has 996 individual features, a principal component analysis (PCA) was applied to reduce the number of dimensions from 996 to 70 before applying t-SNE. The cumulative explained variance of 70 principal components is 0.65. Cumulative explained variance specifies the variance of the dataset explained by each of the principal components [127].

Various methods were tested with the purpose of identifying and explaining clusters in the t-SNE visualization. Initially, metadata such as release date (decade) and popularity were experimented with. However, the variances in the dataset were found to not be explainable by these metadata features alone. In order to make sense of variances and similarities in the dataset, K-means clustering, as well as topic modeling in the form of Latent Dirichlet Allocation (LDA), were applied [10]. The number of K-means clusters was determined by using the “elbow of the curve” in an elbow method as a cutoff point (Figure 4.1a), resulting in 15 clusters. The number of topics and learning decay of the LDA model was based on a grid search performed over these two parameters (Figure 4.1b). The K-means clustering and LDA topic modeling are represented in Figures 4.2 and 4.3.

The most important labels in each LDA topic are displayed in Table 4.2. Looking at the top words for the various topics, some assumptions about the variations in the different clusters can be made. For instance, topic 0 contains the words related to screens and digital items, such as “television”, “monitor”, and “digital_clock”. Topic 2 on the other hand contains terms such as “missile”, “bulletproof_vest”, and “space_shuttle”, indicating that movies within this topic

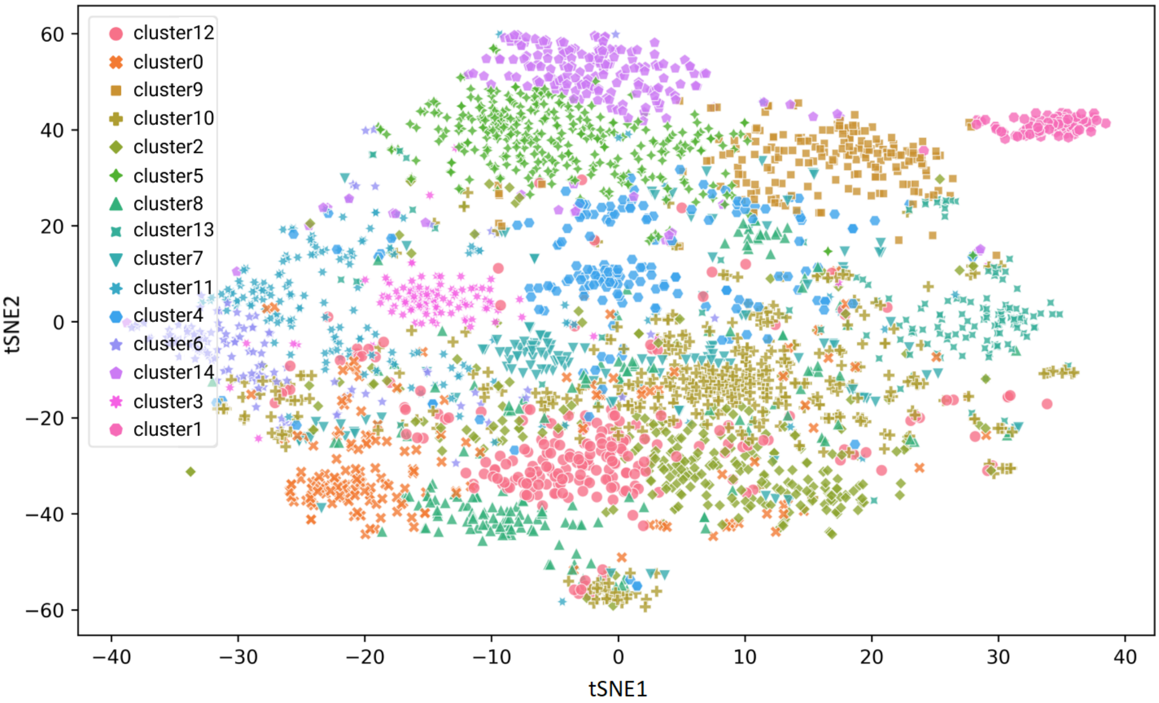


Figure 4.2: t-SNE visualization of the K-Means clustering of DeepCineProp13K

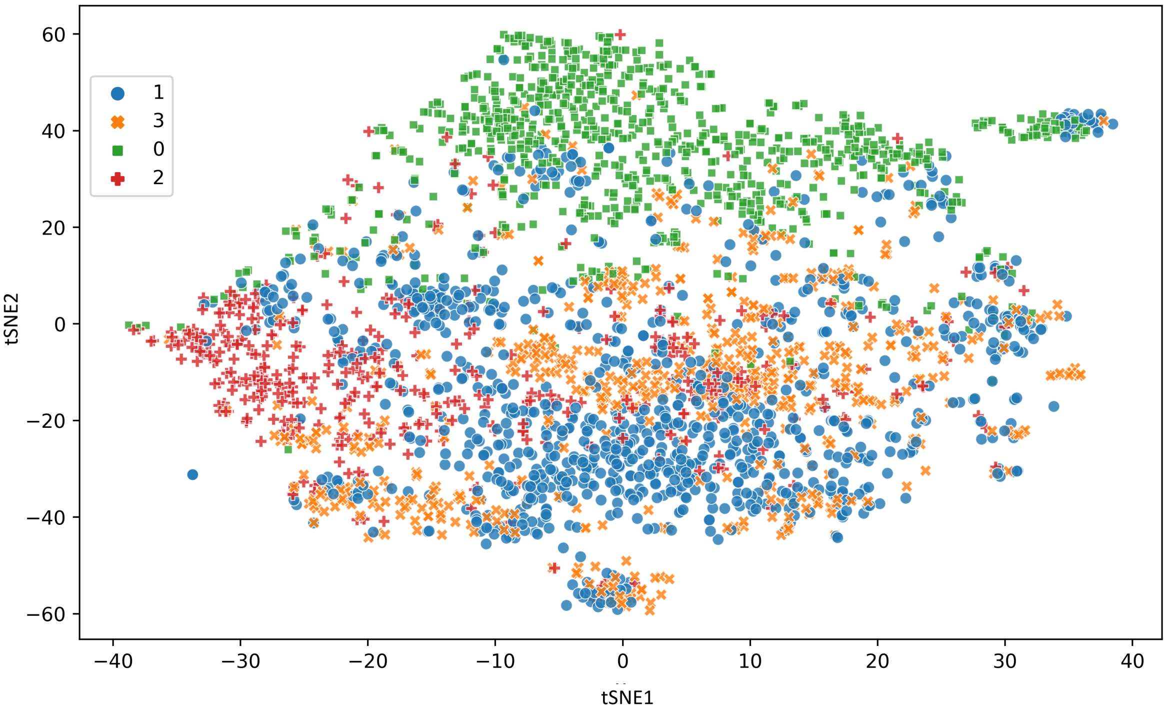


Figure 4.3: t-SNE visualization of the LDA topic modeling of DeepCineProp13K

Cluster	Movie title	Year	IMDB genres	IMDB plot keywords
cluster0	Reservoir Dogs	1992	Crime, Thriller	robbery, gore, gang
	Back to the Future Part II	1989	Adventure, Sci-Fi	year 2015, time machine, delorean
	Back to the Future Part III	1990	Adventure, Sci-Fi	year 1955, time machine, hero
cluster1	Saving Private Ryan	1998	Drama, War	rescue mission, d day, soldier
	RoboCop	1987	Action, Sci-Fi	robot, police, cyborg
	Tarzan	1999	Animation, Adventure	africa, jungle, wild man
cluster2	Star Wars	1977	Adventure, Fantasy	rebellion, jedi, droid
	The Phantom Menace	1999	Adventure, Fantasy	jedi, slavery, galactic war
	Toy Story 2	1999	Animation, Adventure	toy, rescue, sequel
cluster3	Return of the Jedi	1983	Adventure, Fantasy	death star, villain turns good
	Speed	1994	Action, Thriller	bus, bomb, police
	The Devil's Advocate	1997	Drama, Thriller	lawyer, the devil, rape
cluster4	The Shawshank Redemption	1994	Drama	prison, wrongful imprisonment
	The Fellowship of the Ring	2001	Adventure, Fantasy	ring, quest, hobbit
	Titanic	1997	Drama, Romance	iceberg, titanic, mass death
cluster5	Pulp Fiction	1994	Crime, Drama	overdose, neo noir, black comedy
	Mission: Impossible	1996	Action, Thriller	train, betrayal, helicopter
	Die Hard: With a Vengeance	1995	Action, Thriller	good versus evil, cab, helicopter
cluster6	Aliens	1986	Sci-Fi, Thriller	alien, rescue mission, soldier
	Rocky	1976	Drama, Sport	boxer, fistfight, training
	Psycho	1960	Horror, Thriller	motel, shower, money
cluster7	Back to the Future	1985	Adventure, Sci-Fi	time travel, delorean, future
	2001: a Space Oddysey	1968	Sci-Fi	monolith. star child, future shock
	Terminator 2: Judgment Day	1991	Action, Sci-Fi	time travel. future, sequel
cluster8	The Green Mile	1999	Drama, Fantasy	death row, evil, healing
	Total Recall	1990	Action, Sci-Fi	false memory, dystopia, torture
	Grease	1978	Musical, Romance	year 1959, dance contest, singing
cluster9	Raiders of the Lost Ark	1981	Action, Adventure	nazi, archeologist, melting face
	American Pie	1999	Comedy	nudity, prom, sex comedy
	10 Things I Hate About You	1999	Comedy	dating, shrew, prom
cluster10	The Empire Strikes Back	1980	Adventure, Fantasy	famous twist, duel, outer space
	Alien	1979	Horror, Sci-Fi	alien, spaceship, outer space
	Jaws	1975	Adventure, Thriller	shark, beach, monster
cluster11	Batman	1989	Action, Aventure	criminal, maniac, superhero
	Face/off	1997	Action, Sci-Fi	face transplant, villain, maniac
	Contact	1997	Sci-Fi, Thriller	wormhole, religion, nasa
cluster12	Toy Story	1995	Animation, Adventure	toy, rivalry, cowboy
	Mulan	1998	Animation, Adventure	violence, cross dressing
	A Bug's Life	1998	Animation, Adventure	ant, circus, grasshopper
cluster13	American History X	1998	Drama	neo nazi, prison, hatred
	Blade Runner	1982	Sci-Fi, Thriller	tech noir, cyberpunk, dystopia
	The Rock	1996	Action, Thriller	prison, escape, bomb
cluster14	American Beauty	1999	Drama	midlife crisis, unfaithful wife
	Home Alone 2: Lost in Ne...	1992	Comedy, Family	christmas movie, wish fulfillment
	Willy Wonka & the Choco...	1971	Fantasy, Family	greed, prize, wish fulfillment

Table 4.1: The three most popular movies in each of the K-means clusters.

Topic 0	Topic 1	Topic 2	Topic 3
television	suit	window_screen	book_jacket
monitor	lab_coat	prison	butcher_shop
web_site	barbershop	geyser	balloon
envelope	prison	space_shuttle	stage
cash_machine	abaya	missile	suit
digital_clock	punching_bag	bulletproof_vest	jellyfish

Table 4.2: Top keywords in the LDA topics.

contain objects related to war and science fiction.

In order to explore the K-means clusters, the most popular movies from each cluster were checked. Table 4.1 displays all clusters and information linked to the corresponding three most popular movies. The information include title, year of release, movie genre collected from IMDB[1], as well as plot keywords also collected from IMDB. Looking at the movies in each cluster, it is noticeable that most clusters contain movies similar either in genre and/or plot keywords. For instance, the most popular movies in cluster12 are all children's animation movies produced by Disney. Examples of clusters showing some degree of thematic and semantic correlation include cluster7 which has Sci-Fi movies that involve time travel of some sort, while two of the three most popular movies in cluster14 are Familiy movies about wish fulfillment.

Even though some correlation either in style or semantics is apparent in most clusters, not all movies in each cluster seem to belong together. In cluster8, neither of the three most popular movies share genre, nor do they share any plot keywords. The same goes for the movies in cluster1. Movies belonging to the same franchise, such as *Star Wars* and *Back to The Future*, represents another intuitive relation for grouping movies. Accordingly, *Back to the Future Part II* and *Back to the Future Part III* are both in cluster0. However, the original movie in the franchise, *Back to the Future*, is in cluster7. The case for the *Star Wars* movies is similar, with two movies in cluster2, one in cluster3, and one in cluster10.

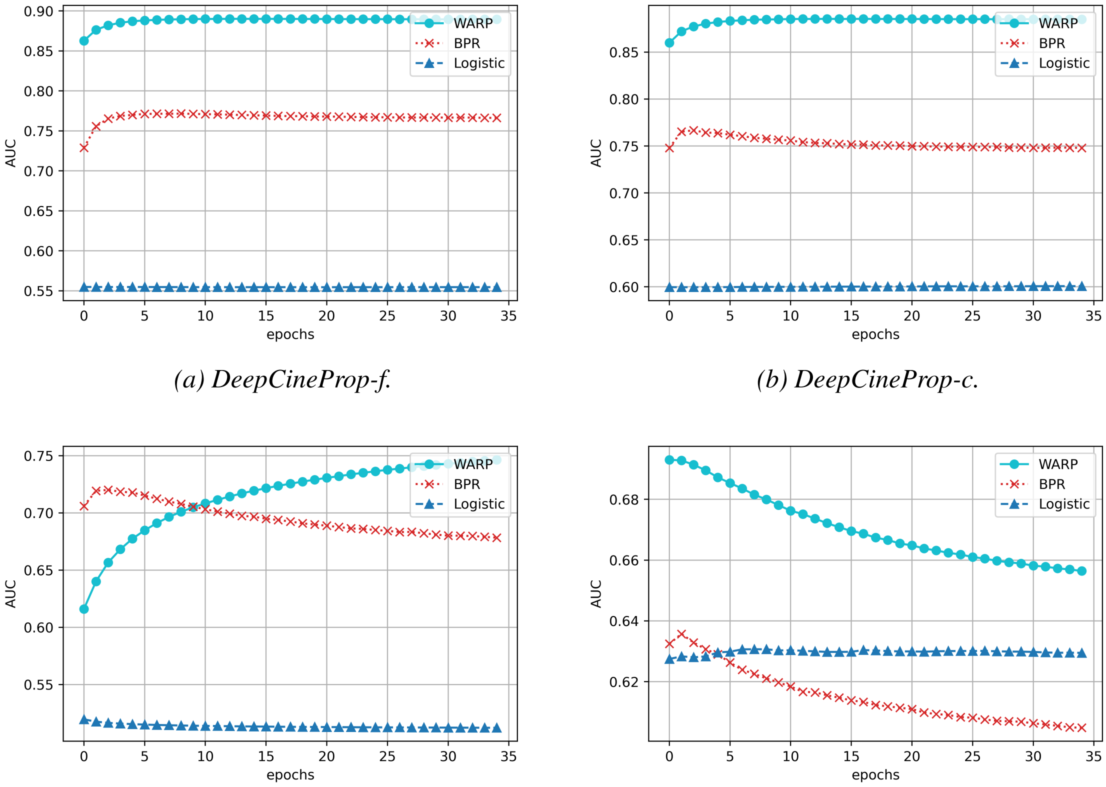
While the results from this exploratory analysis of DeepCineProp13K reveal potential weakness for grouping certain types of movies and themes, the overall results are promising. The K-means clustering demonstrates capabilities of separating movies according to both stylistic and semantic features.

4.2 Experiment B: Recommendation Quality

In the second set of experiments, the quality of the recommendation based on automatic visual features, extracted by the deep learning model were measured. Figure 4.3 represents the results obtained in this experiment.

First of all, as it can be seen, both versions of the proposed recommendation technique

[1] imdb.com/



(a) DeepCineProp-f.

(b) DeepCineProp-c.

(c) Tag.

(d) Genre.

Figure 4.4: AUC over epochs with different loss functions trained with DeepCineProp-f, DeepCineProp-c, tag, and genre.

Feature	Type	Precision@K	Recall@K	AUC	Reciprocal Rank
Genre	*manual*	0.008	0.007	0.661	0.035
Tag	*manual*	0.053	0.068	0.721	0.147
DeepCineProp-c	*automatic*	0.116	0.123	0.885	0.270
DeepCineProp-f	*automatic*	0.122	0.123	0.890	0.282
CineSub	*automatic*	**0.177**	**0.172**	**0.962**	**0.381**

Table 4.3: Comparison of the recommendation quality based on automatic features and manual features.

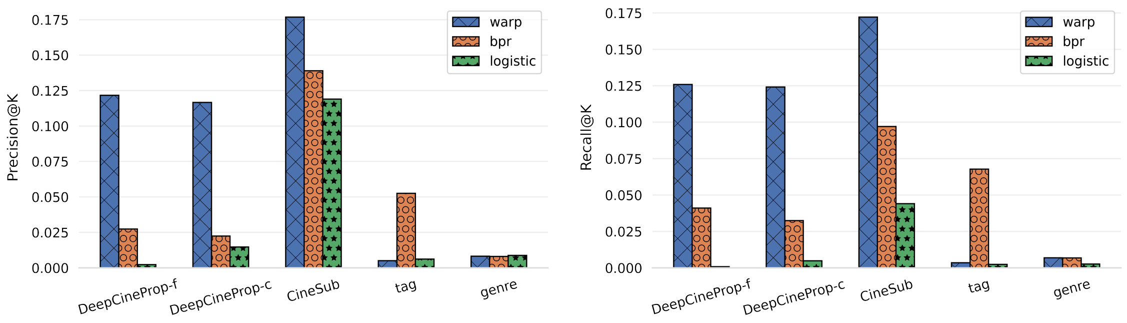
(DeepCineProp-f and DeepCineProp-c) — based on visual features — outperform both the genre and tag baselines. However, the CineSub baseline is not beaten in any of the measures. In terms of Precision@K, DeepCineProp-f and DeepCineProp-c respectively achieve scores of 0.122 and 0.116. The best precision score is obtained by recommendation based on CineSub with a score of 0.177, whereas recommendation based on manual features, i.e., genre and tag, received the lowest scores, i.e., 0.008 and 0.053, respectively. In terms of Recall@K, similarly, both DeepCineProp-f and DeepCineProp-c achieved better results than genre and tag with the scores of 0.126 and 0.123, respectively. The best performance has been observed for recommendation based on CineSub with a score of 0.172. The recommendation based on genre and tag have performed the worst with the scores of 0.007 and 0.068, respectively.

In terms of AUC, recommendation based on DeepCineProp-f has achieved a great score of 0.890, however, CineSub still has obtained the best score of 0.962. Recommendation based on DeepCineProp-c has obtained the third best result with a score of 0.885. Recommendation based on genre and tag have received the lowest scores, i.e., 0.661 and 0.721, respectively. Finally, in terms of Reciprocal Rank, again, proposed recommendation techniques based on either DeepCineProp-f and DeepCineProp-c have achieved higher scores than genre and tag. While the respective observed scores for DeepCineProp-f and DeepCineProp-c were 0.282 and 0.270, CineSub achieved 0.381. The scores observed for recommendation based on manual features were significantly lower. Both genre and tag have shown the worst performances with respective scores of 0.035 and 0.147.

4.3 Experiment C: Comparing Loss Functions

In the third set of experiments, I have compared the recommendation based on automatic features when different types of optimization algorithms are used. The results are illustrated in Figure 4.5 and 4.6.

First of all, as it can be seen, different loss function (hence optimization algorithm) can yield different recommendation quality for each type of automatic features. For the visual features — either DeepCineProp-c or DeepCineProp-f — the best results have been achieved



(a) Precision

(b) Recall

Figure 4.5: Comparison of recommendation based on automatic features using different optimization methods in terms of (left) Precision and (right) Recall

using *warp* loss function with consideration to all metrics, i.e., Precision@K, Recall@K, AUC, and Reciprocal Rank. Surprisingly, *bpr* loss function does not perform well compared with warp, and in some cases (e.g., Precision), it yields less than 1/3 of the score. The worst performance for both DeepCineProp datasets is displayed by the logistic loss function, which is outperformed by warp and bpr across all metrics.

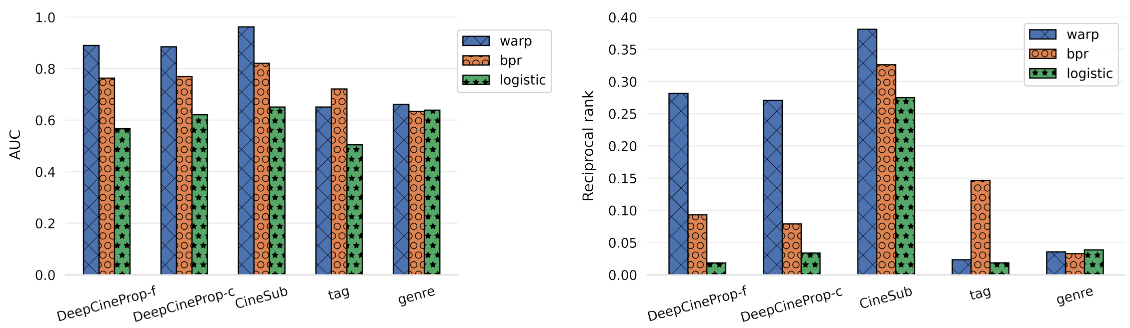
For the CineSub features, the best results are still achieved by the warp loss function for all metrics. In contrary to DeepCineProp-f and DeepCineProp-c, bpr nearly performs as well as warp on metrics such as Reciprocal Rank. The worst results are again obtained by the logistic loss function. Since the features of the DeepCineProp datasets and CineSub are of the same categorical type, we could expect that the same loss function would achieve superior performance across all three datasets. These similarities in results can serve as confirmation that these datasets share similarities in their nature.

Comparing the loss functions for the manual features tag and genre shows different results than the automatic features described above. For tag features, bpr achieves the highest results across all metrics, whereas warp and logistic loss both yield far inferior results on metrics such as Precision and Recall. The results for genre show that all loss functions display fairly similar performance.

Overall, these promising results have shown the excellent performance of hybrid recommendation based on visual features, using different optimization methods. The results have clearly illustrated the substantial potential behind these features that can be exploited when no other types of content features are provided to a movie recommender system.

4.4 Experiment D: Real-User study

The fourth and final experiment utilizes the developed web application to let real users evaluate recommendation quality for three of the implemented models along the metrics described in section 3.5.2. Personality and demographic effects on the preferences are considered. Additionally, the usability of the web application itself is assessed, using the System Usability Scale



(a) AUC *(b) Reciprocal rank*

Figure 4.6: Comparison of recommendation based on different automatic features using different optimization methods in terms of (left) AUC and (right) Reciprocal Rank

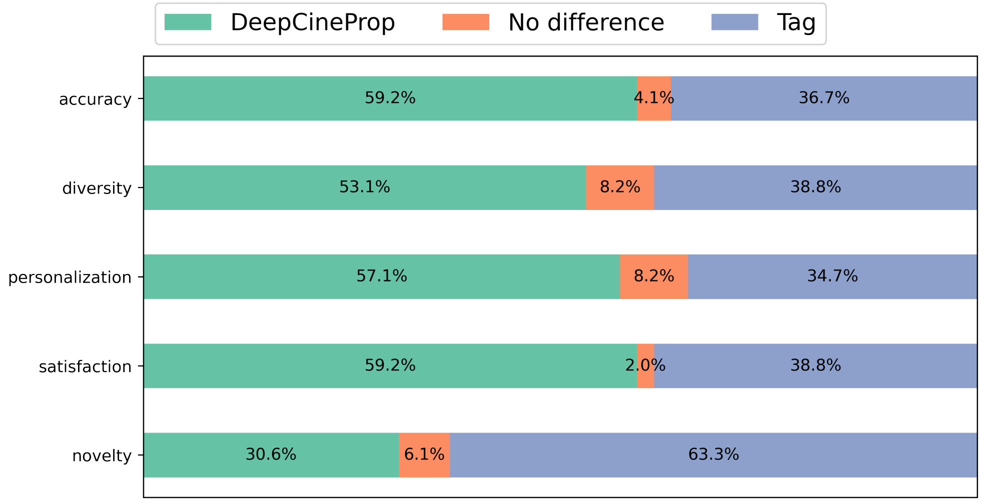


Figure 4.7: Summary of the results for the user evaluation of DeepCineProp vs. Tag.

(SUS). An overview of the collected recommendation quality evaluation data is provided in Table 4.4, while 4.14 represents the SUS responses.

Recommendation Quality

The recommendation quality of the recommenders is evaluated as described in Section 3.5.2. Table 4.4 gives an overview of the responses, while the correlations between the recommendation quality factors are represented in Figure 4.10. In total, 762 ratings for 278 unique movies were given by the 150 participants. In 28% of the cases, users would watch the trailer before rating a movie. The average completion time of the full study was 11.5 minutes. The DeepCineProp, CineSub, and tag-based recommenders were evaluated by 103, 101, and 96 users respectively. A total of 391 unique movies were recommended, with DeepCineProp and CineSub providing 195 and 197 unique recommendations respectively. The tag recommender provided 118 unique recommendations. A detailed explanation of the results is given in the

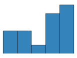
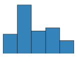
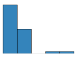
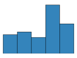
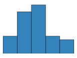
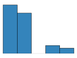
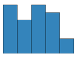
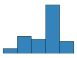
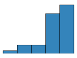
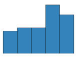
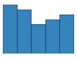
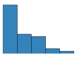
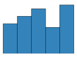
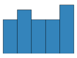
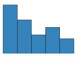
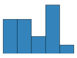
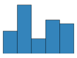
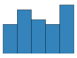
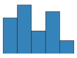
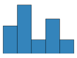
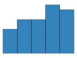
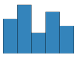
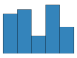
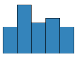
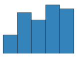
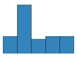
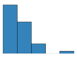
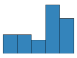
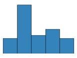
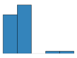
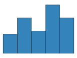
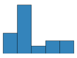
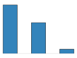
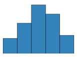
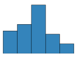
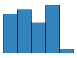
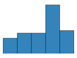
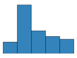
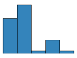
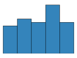
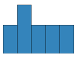
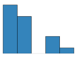
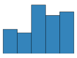
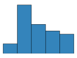
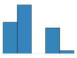
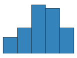
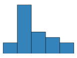
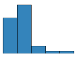
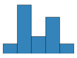
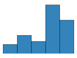
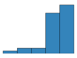
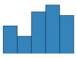
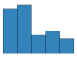
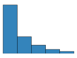
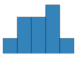
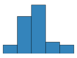
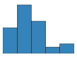
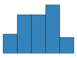
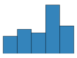
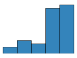
Question: Which list...	DeepCP v. CSub	DeepCP v. tags	CSub v. tags
Accuracy			
1. has more movies that you find appealing?			
2. has more movies that might be among the best movies you see in the next year?			
3. has more obviously bad movie recommendations for you?			
4. does a better job of putting better movies at the top?			
Diversity			
5. has more movies that are similar to each other?			
6. has a more varied selection of movies?			
7. has movies that match a wider variety of moods?			
8. would suit a broader set of tastes?			
Personalization			
9. better understands your taste in movies?			
10. would you trust more to provide you with recommendations?			
11. seems more personalized to your movie ratings?			
12. more represents mainstream tastes instead of your own?			
Satisfaction			
13. would better help you find movies to watch?			
14. would you be more likely to recommend to your friends?			
15. of recommendations do you find more valuable?			
16. would better help to pick satisfactory movies?			
Novelty			
17. has more movies you do not expect?			
18. has more movies that are familiar to you?			
19. has more pleasantly surprising movies?			
20. has more movies you would not have thought to consider?			

Table 4.4: Results from each question of the recommendation quality evaluation with real users (N=150). Some abbreviations are used to make the dataset names fit in the table, including DeepCP (DeepCineProp-f) and CSub (CineSub).

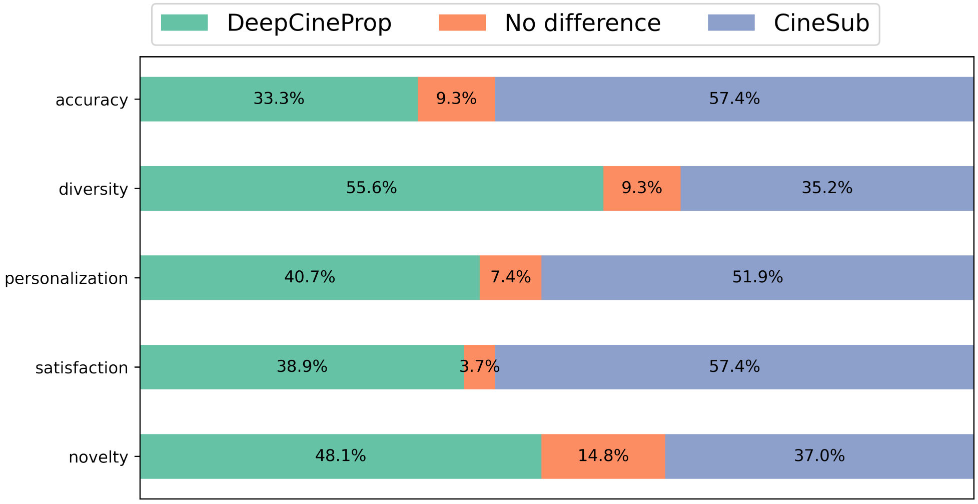


Figure 4.8: Summary of the results for the user evaluation of DeepCineProp vs. CineSub.

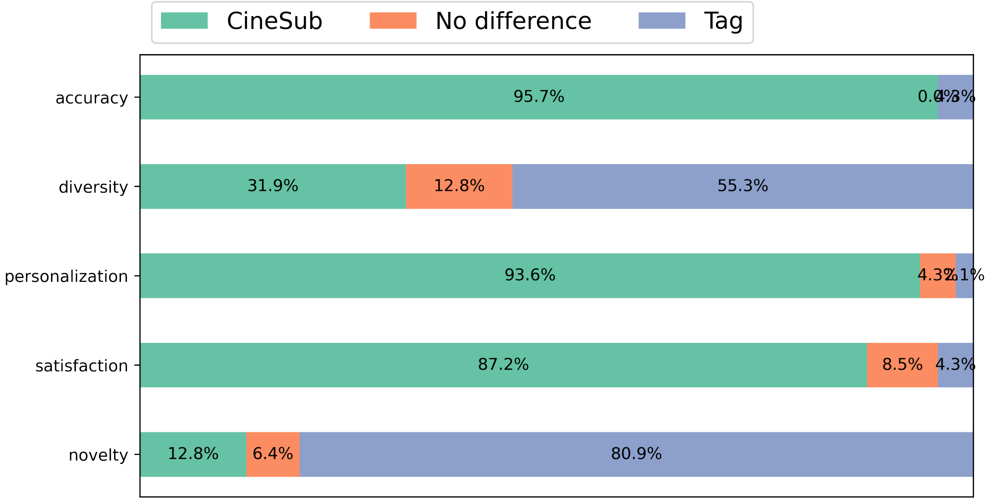


Figure 4.9: Summary of the results for the user evaluation of CineSub vs. Tag.

following:

Accuracy. The results from questions that address the perceived accuracy of the recommender lists show that both the DeepCineProp and CineSub recommenders perform significantly better than the tag baseline. The fraction of participants perceiving the personalization of recommendations given by DeepCineProp as better than tag is 61.2%, while 6.1% thought they performed equally in this regard. Most notably, only 4.2% of participants think that tag gives more personalized recommendations than CineSub. While the margin is smaller when compared with DeepCineProp, CineSub still arise as the model that provides more personalized recommendations for 59.2% of users.

Diversity. The only orthogonal factor in the recommendation quality evaluation (Figure 4.10) is diversity. This result differs from Ekstrand et al. [38], where diversity was found to have significant positive correlations with both accuracy and satisfaction. In this category, the performance of both tag and CineSub baselines are lower than DeepCineProp, which emerges as the more diverse of the recommenders. In the comparison between tag and CineSub, 55.3% choose tag as more diverse, while 44.7% find CineSub equally or more diverse in its recommendations. DeepCineProp performs better than both baselines, and is considered to give equally or more diverse recommendations than CineSub and tag among 64.8% and 63% of participants respectively.

Personalization. Similar to the accuracy factor, DeepCineProp and CineSub both outperform the tag-based recommender in perceived personalization. Looking at the Pearson coefficients in Figure 4.10, there is indeed a clear positive correlation between accuracy and perceived personalization ($p < 0.001$). With tag as baseline, 57.1% of participants perceive the DeepCineProp recommender as more personalized, whereas CineSub achieves a higher level of personalization for 93.6% on the same baseline. However, when comparing DeepCineProp and CineSub directly, there is minimal difference between the two. Head-to-head, CineSub gets a higher score than DeepCineProp for perceived personalization for 51.9% of the participants, while 48.1% say DeepCineProp gave equally or more personalized recommendations.

Satisfaction. The positive correlation of the satisfaction factor to both accuracy (*coeff.* 0.88, $p < 0.001$) and personalization (*coeff.* 0.84, $p < 0.001$) is apparent in the results, seeing that DeepCineProp and CineSub are given higher satisfaction ratings compared with the tag baseline. CineSub is perceived as having the best capabilities in giving satisfying recommendations by 57.4% of users in comparison with DeepCineProp.

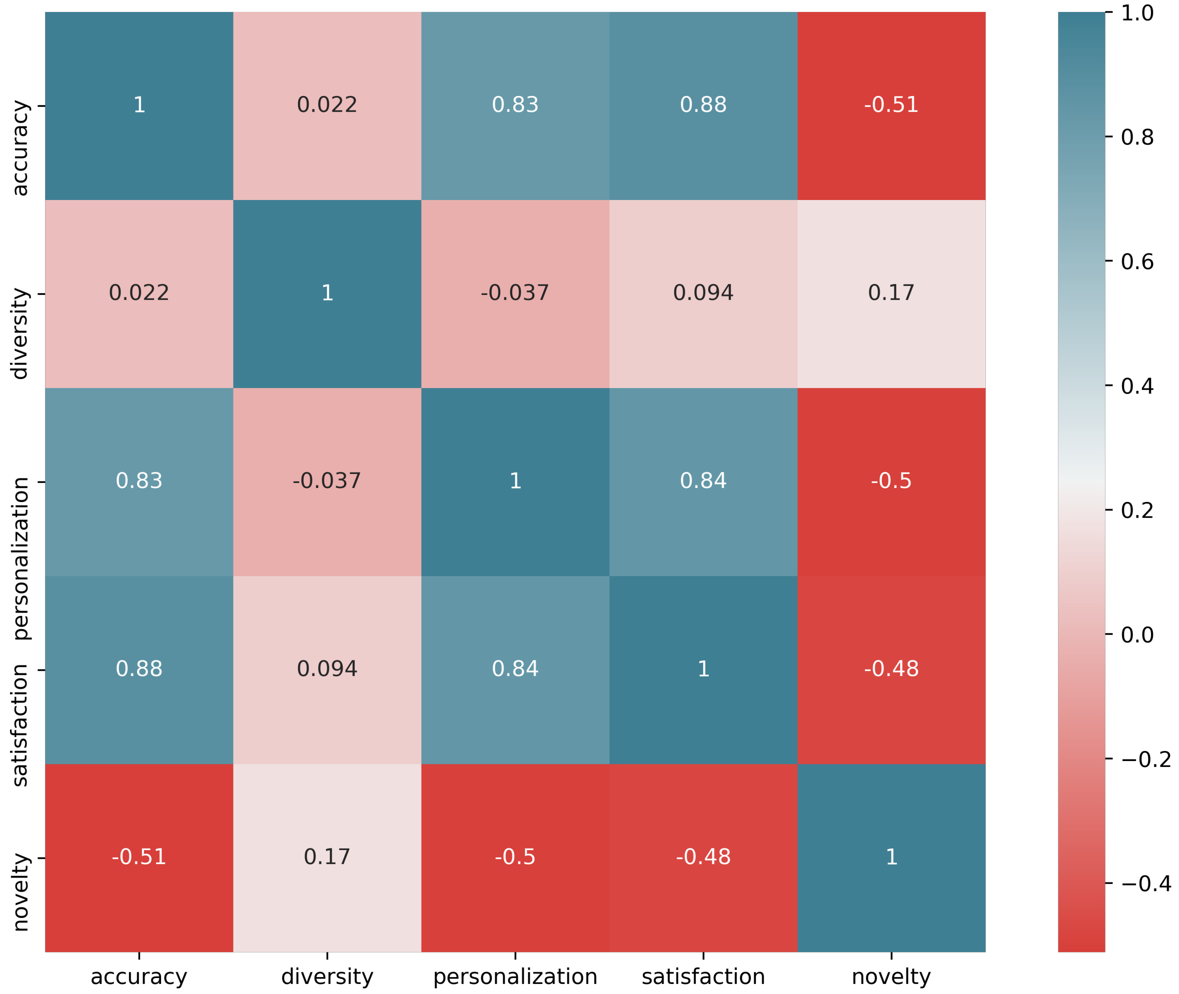


Figure 4.10: Correlations between the different recommendation quality factors.

Novelty. The perceived capability of recommending novel items to the user is negatively correlated with the factors accuracy (*coeff.* -0.51, $p < 0.001$), personalization (*coeff.* -0.5, $p < 0.001$), and satisfaction (*coeff.* -0.48, $p < 0.001$). Being outperformed in each of these three metrics, the tag model unsurprisingly achieves a higher score for novelty than the two other recommender models. DeepCineProp is viewed to give more novel recommendations than CineSub. However, compared with the tag baseline, DeepCineProp and CineSub achieve higher novelty among 30.6% and 12.8% participants respectively.

Personality and demographic influences.

To get a better understanding of what influences a participant's choice of preferred recommender model, the demographic and personality responses were assessed.

The demographic and media consumption distribution of the participants is displayed in

Table 4.5.

K-means clustering was adopted to investigate the preferences of users with different personality types. The elbow method was adopted to identify the right number of clusters. The results, displayed in Figure 4.11, indicate that 3 is the best number of clusters to divide users into based on their personality. In order to visualize the clusters, PCA was applied, resulting in the plots shown in Figure 4.12. Some general observations about the characteristics of users in the different clusters (Table 4.13) include that users in Cluster 0 tend to score low on neuroticism, meaning they are emotionally stable. Users in Cluster 0 also generally score lower on conscientiousness than the other clusters. Participants in Cluster 1 tend to have highly conscientious and neurotic personalities, while those in Cluster 2 often are introverted as well as conscientious.

Comparisons between the preferences of users in different personality clusters show some differences. In particular the preferences of participants from Cluster 0 distinguish themselves from the two other clusters. When comparing DeepCineProp with the tag baseline, 80% of participants in Cluster 0 find CineProp more satisfying, while the same numbers for Cluster 1 and Cluster 2 are only 50% and 59% respectively. The same pattern is apparent when DeepCineProp is compared with CineSub. Although most users found CineSub to give more satisfying results overall, 50% of participants in Cluster 0 find DeepCineProp to give higher satisfaction. Any significant differences between the preferences of users in Cluster 1 and Cluster 2 were not observed.

Although no particular differences were observed from users of different demographic groups, a few effects of media consumption on user preferences were observed. Participants who watched more movies had a higher tendency of choosing tag over DeepCineProp for satisfaction ($coeff. = 0.32$, $p = 0.026$) as well as tag over CineSub for accuracy ($coeff. = 0.34$. $p = 0.02$). Those who spent more time watching movies also more often saw recommendations produced by CineSub as less diverse than those of DeepCineProp ($coeff. = -0.32$. $p = 0.018$). Since significant correlations of media consumption and preferences are few and far between, media consumption patterns in general cannot be concluded to explain preferences toward either of the evaluated recommender models.

Usability.

To summarize the subjective user experience, the overall SUS score was calculated based on the collected responses (N=145). The system achieved a mean SUS score of 77.3. A baseline of 68 is typically used, which is the average SUS score over a large number of usability studies, reported in Sauro [108]. The achieved score can therefore be said to be above average. According to the curved grading scale interpretation of SUS scores proposed by Sauro and Lewis [109], the score is within the range of a B+, or a percentile rank of 80%. The SUS responses are represented in Figure 4.14 as a raincloud plot, which combines jittered raw data points, split-half violin plot, mean, and boxplot [4].

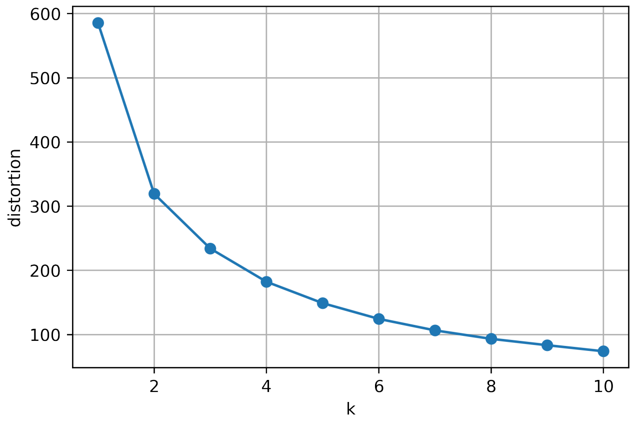


Figure 4.11: Elbow method for identifying participant personality clusters

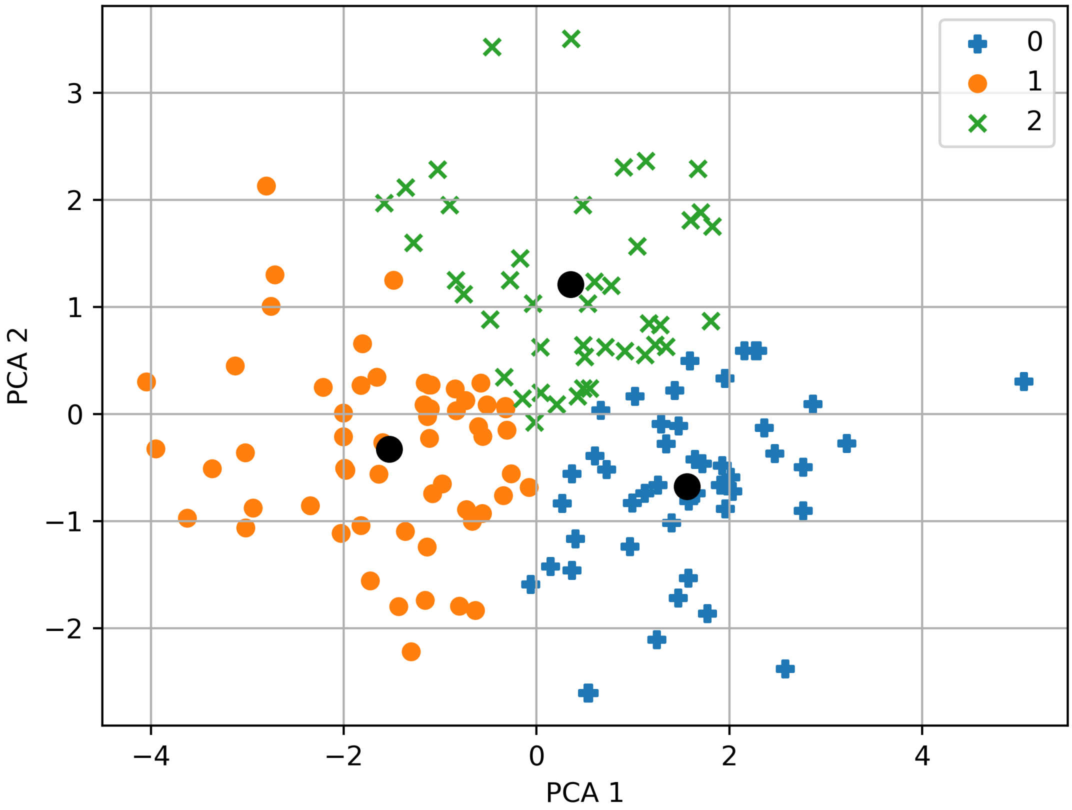
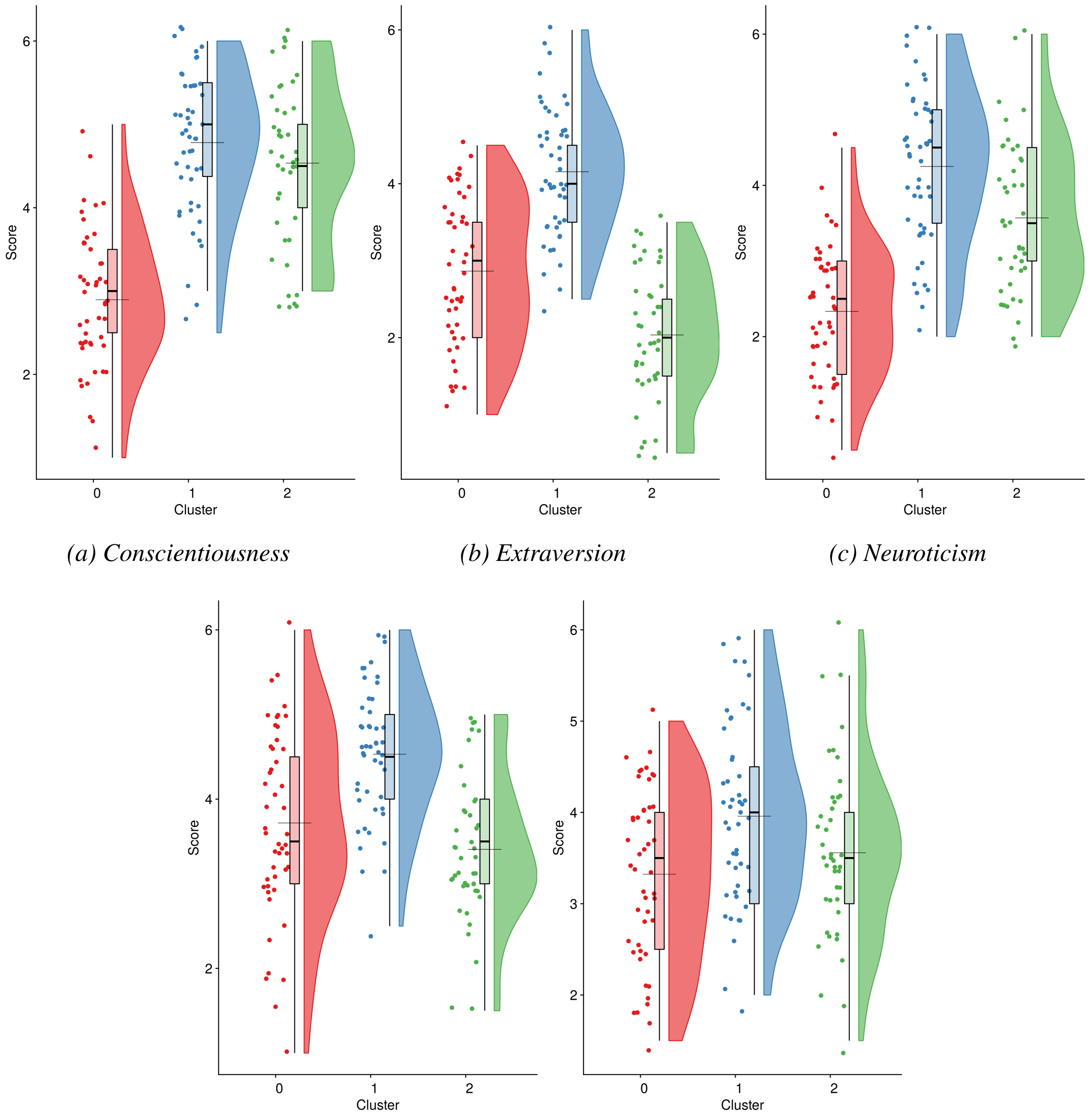


Figure 4.12: K-means clusters of participants' personalities.



(a) Conscientiousness *(b) Extraversion* *(c) Neuroticism*

(d) Openness *(e) Agreeableness*

Figure 4.13: Comparison of personality traits for the different personality clusters.

Demographics	*Age*	18-24 71	25-30 39	31-40 30	41+ 10
	Gender	Female 51	Male 97	Other 2	
	Native English-speaker	Yes 46	No 104		
Media consumption	*TV series (hrs per week)*	<1 13	1-4 38	5-10 63	>10 36
	Movies (hrs per week)	<1 5	1-2 43	3-6 73	>6 29
	Cinema (times per year)	<1 20	1-2 41	3-6 46	>6 43

Table 4.5: Participant characteristics

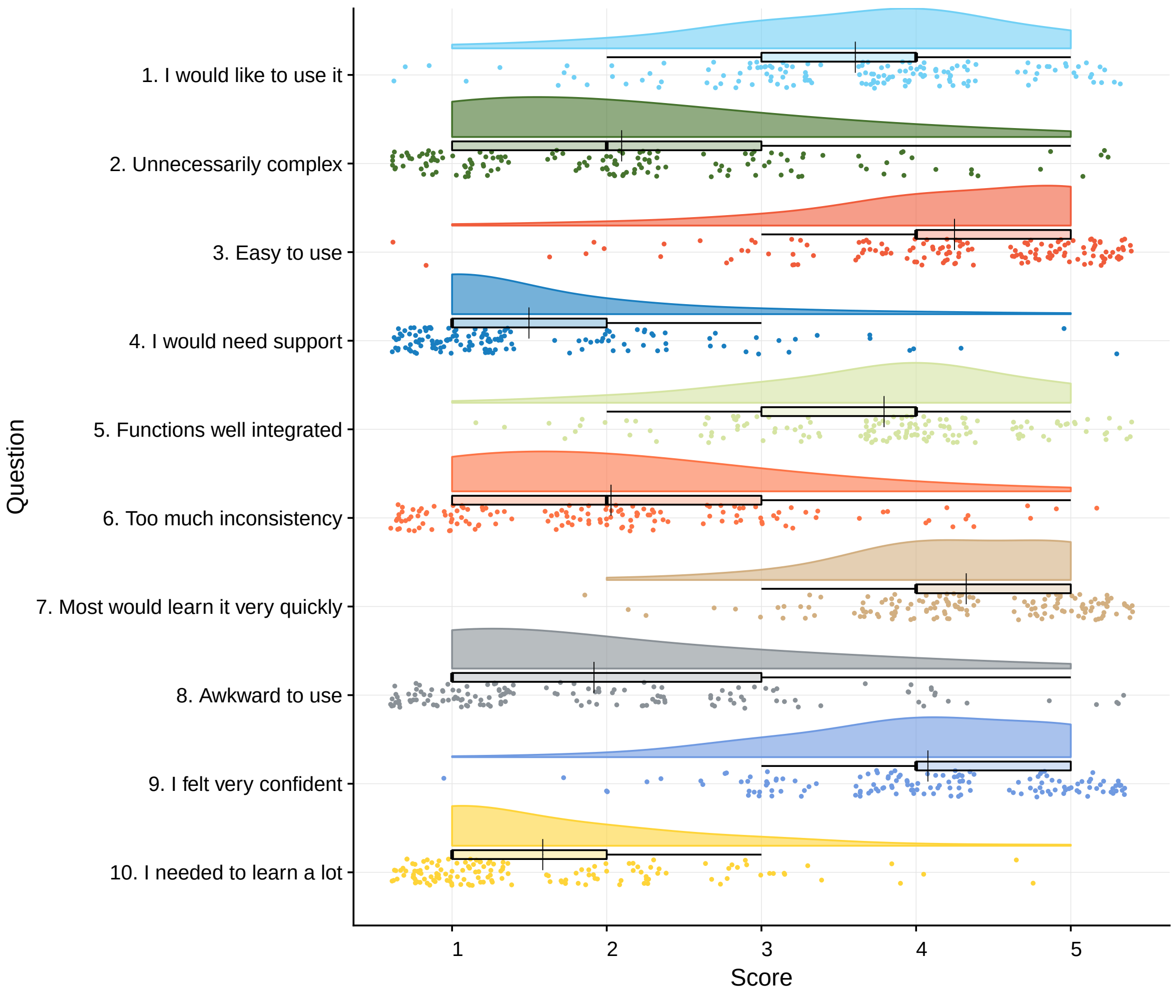


Figure 4.14: System Usability Scale results (1="Strongly disagree", 5="Strongly agree"). The questions on the y axis are short versions of the questions originally asked.

The Sauro [108] general SUS benchmark of 68 does not take into account that perceived usability differs significantly in different types of products and interfaces. To be able to better interpret the SUS score, we can compare it to more specific categories related to the system. According to categorical data collected by Sauro [108], public-facing websites receive an average score of 67 which would give a C on the curved grading scale. Even though our system is not a native application, it is designed to work for mobile interfaces. It can therefore be interesting to note that on average, 15 of the most popular mobile applications achieved a total SUS score of 77.7, which is fairly similar to the result of 77.3 which was achieved in this study [108].

The adjective rating given by users was positively correlated with the SUS scores, with a Pearson coefficient of 0.64 ($p < 0.01$). The mean adjective rating across all users was 5.5 out of a maximum of 7, which puts the system between “good” and “excellent” in overall user-friendliness. These results correspond well with the findings in Bangor, Kortum and Miller [7], which display that the mean SUS score of a system with “good” adjective rating is 71.4, while the mean SUS score is 85.5 for systems that achieve “Excellent” adjective rating.

Additionally, the effect of demographics and media consumption on the given SUS scores and adjective ratings by users was investigated. Age group had a slight positive correlation with SUS score with a coefficient of 0.145 ($p = 0.082$). Since the p-value is more than 0.05, this effect is discarded as not statistically significant, and it is therefore not certain that older participants are more inclined to find the system more usable. When it comes to personality, participants in cluster 2 (Figure 4.12) give the system an overall higher rating than participants in cluster0 and cluster1. While the mean SUS scores from users in cluster 0 and cluster 1 are 74.3 and 75.7 respectively, users in cluster 2 gives the system an average score of 82.6. This indicates that personality traits may affect how a user perceives the usability of a recommender system interface.

According to the results presented, the system achieves a more than acceptable level of subjective usability among participants. An achieved score of 77.3, which is well above the general average and within the percentile rank of 80%, demonstrates that it provides a user-friendly experience for participants. Considering that it is a prototype framework, these are promising results for further development.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis, a hybrid technique to generate recommendation based on visual features automatically extracted from movies is proposed.

The cold start problem for movie recommendation — more specifically the new item problem — has been addressed by extracting novel features for movie recommendation and comparing the recommendation capabilities of these with more traditional content features. The research was carried out by following a four-step procedure which included:

1. Establishing the research context and state-of-the-art by conducting a literature review (detailed in Chapter 2).
2. Forming a novel dataset of visual features from 12,875 movie trailers with the use of deep learning, as well as a subtitle-based dataset from 3,405 movies (explained in Chapter 3).
3. Evaluation of accumulated datasets in offline experiments to investigate and compare the recommendation capabilities of these (described in Chapter 4).
4. Integration of the best-performing recommender models in a real recommender system and evaluation of these in online experiments with 150 real users (described in Chapter 4).

5.2 Main contributions

This thesis advances the state-of-the-art of content-based movie recommender systems through the contributions listed in the following:

- *Proposing a novel hybrid recommendation technique based on visual features automatically extracted with deep learning:* In Chapter 3, approaches utilizing classification label output of deep visual features from the key frames of 12,875 movie trailers are proposed. One approach includes taking advantage of the confidence output of each label, while another uses the popular TF-IDF method to aggregate the features. The final datasets are made openly available online through the project's Github repository.

- *A comprehensive evaluation of proposed recommendation approach in both offline and online experiments, including consideration of different optimization methods and comparisons with different baselines on various evaluation metrics:* Chapter 3 describes the methodology used in the offline experiments, while results of these are presented in Chapter 4. The offline evaluation experiments include exploratory analysis, recommendation quality, and comparison of different loss functions. In the exploratory analysis, clusters in the DeepCineProp13K dataset are recognized and explored by the use of K-means clustering and LDA topic modelling. The dataset is visualized by using the dimensionality reduction techniques PCA and T-SNE. In the recommendation quality experiment, DeepCineProp-based recommenders are compared with baselines based on traditional content features, such as tag and genre, as well as novel features from the proposed CineSub dataset. The comparison experiment of loss functions investigates how recommender models trained with different content features perform with different optimization methods, including bpr, warp, and logistic loss functions.

- *Collecting a large dataset of subtitles from 3,405 full length movies and exploiting them in a baseline recommendation technique:* The collection and aggregation of subtitle features are described in Chapter 3. In Chapter 4, the subtitle dataset, named CineSub3K, is used as baseline model for the DeepCineProp-based recommenders in both offline and online experiments. CineSub 3K is made openly available in the project's Github repository.

- *Developing a framework for evaluating and comparing different movie recommenders with real users as a modern web application, including an evaluation of the framework's usability:* The design process, specifications, and properties of the movie recommender evaluation framework, named Samvise, are described in Chapter 3. In Chapter 4, the framework is successfully utilized for the online experiment with real users and evaluated for usability by 145 users with the System Usability Scale.

5.3 Conclusion

The results of the offline evaluation of recommendation quality as detailed in Section 4.2 have shown that models trained on the proposed types of visual features achieve higher scores on algorithmic metrics compared with more traditional content features. The results from the online evaluation with 150 participants, as detailed in Section 4.4, substantiate the offline results. Participants find the DeepCineProp-based recommender to provide more accurate, personalized, and satisfying results than the tag-based recommender. Additionally, tag, which represents more traditional content features, produces recommendations with higher novelty, DeepCineProp is found to give diverse recommendations to a higher extent. As a conclusion of RQ 1.1, the findings demonstrate that the proposed recommendation technique with visual features extracted with deep learning provide better recommendation quality compared with more traditional recommendation approaches that use manually created metadata.

When comparing the proposed DeepCineProp recommender on algorithmic performance metrics against CineSub, both of which are based on features that can be extracted automatically, CineSub achieves higher scores. The offline evaluation results are again in line with the online evaluation in regards to accuracy and satisfaction, as participants find that the CineSub recommender provides more accurate and satisfactory recommendations than DeepCineProp. However, DeepCineProp still achieves better results than CineSub in producing diverse and novel recommendations for users. Moreover, the two recommenders perform fairly evenly when it comes to perceived personalization of recommendations. According to the presented findings of this thesis, the response to RQ 1.2 is that the proposed recommendation technique based on visual features provides less accurate and satisfactory recommendations than subtitle features, but performs equally good or better in terms of personalization, novelty, and diversity. It should also be noted that the dimentionality of CineSub should be reduced in order to decrease computing time. Such refining of the dataset may negatively affect the performance in terms of recommendation quality and potentially even the gap between DeepCineProp and CineSub.

In terms of RQ 2.1, the presented findings in sections 4.2 and 4.4 show high similarity between the outcome of algorithmic performance metrics and perceived accuracy, satisfaction, and personalization of recommendation lists among real users. As expected from previous studies, the novelty metric had a negative correlation to accuracy and satisfaction. Perceived diversity, on the other hand, was found to be an orthogonal factor in the online evaluation, which contradicts results of previous research [38].

With regard to RQ 2.2, some user characteristics were found to affect user preferences. The results from the online evaluation show that participants with low conscientiousness who also are emotionally stable tend to have a higher preference towards the recommendations provided by DeepCineProp. Other factors were also assessed, including media consumption and demography. However, none of these were found to explain user preferences towards either of the evaluated recommender models.

In the context of RQ 2.3, the prototype framework achieved usability scores well above general average in the SUS evaluation (Section 4.4). The findings of the usability assessment did not show any correlation between media consumption habits or demography to how users perceived the usability of the system. However, it was found that users who shared certain personality traits would be more happy with the usability than users with other personalities. The results of the usability assessment demonstrate that the framework provides a user-friendly experience for participants of a movie recommender system evaluation with recommendation based on visual features.

5.4 Limitations and Future Work

The research problems and the approach of this thesis include limitations as well as possibilities for further research to be carried out. This includes work on feature extraction for DeepCineProp13K as well as the proposed subtitle-based CineSub3K. It also includes further development of recommendation output, as well as evaluation with different baselines.

One benefit of utilizing crowdsourcing platforms such as Prolific or Mturk, as opposed to voluntary users recruited by the use of social media, is that we get “blind” users. These are users that have no special connection with the researcher, the experiment, or the system [111]. With “blind” users, we avoid biases such as potentially more predictive behavior due to familiarity with the experiment or system, or socially desirable answers as a consequence of friends or colleagues wanting to please the researcher [90, 117]. However, paid workers from crowdsourcing platforms are more likely to cheat when performing assigned tasks [36]. When comparing with participants from other crowdsourcing platforms, such as Amazon’s Mturk[1], participants from Prolific have shown to be less dishonest, in addition to being more diverse and naive [94]. Prolific also benefits from a high grade of transparency for both participants and researchers [91].

Although quality of responses from crowdsourcing platforms like MTurk and the utilized Prolific is generally high, the online evaluation was not implemented on a real video streaming service. This may be a limitation to the evaluation, as the participants’ intentions of participating are not about finding something to watch, as it would be in a recommender system on a video streaming site.

The CNN model for visual feature extraction is only trained on recognizing objects from the ImageNet dataset. A priority for future work would be to complement the data in the DeepCineProp13K dataset by utlizing a model trained on a different dataset to go beyond the current implementation and recognize other types of features (i.e. actions, affective dimensions, scenery).

The CineSub3K dataset should be refined further as to increase computing time for training a recommendation model with it. Being as high-dimentional as it is now, the achieved recommendation performance may not be representative for recommender systems using a refined version to implement CineSub3K on a larger scale. Testing this recommendation approach on videos and movies with subtitles in other languages than English also represents an exciting problem to research in the future.

In addition to general content-based recommendation, subtitles can be utilized to determine moods and sentiments in a movie. Such approach would be relevant in context-aware recommender systems, which utilize contextual information, i.e. mood, to provide recommendations

[1] mturk.com/

to users.

Since the features of both DeepCineProp13K and CineSub3K are represented by labels, it is feasible to implement recommender systems that utilizes either of these datasets which provide explanations to a user. A future research opportunity lies in evaluating explanations based on these two datasets with existing content-based style explanation techniques as baselines.

In this thesis, the final layer of image classification labels for key-frames generated by the CNN model is used to train the recommender system. In addition to the dataset of labels, the project's Github repository contains a dataset of features for the same movies, but without the last classification layer. The fully connected layer consists of a 4096-dimensional feature vector of each input image instead of the 1000-dimensional label feature vector. Both of these datasets are made openly available for further exploration and research.

Bibliography

[1] Abdel-Hamid, Ossama, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn and Dong Yu. 2014. "Convolutional neural networks for speech recognition." *IEEE/ACM Transactions on audio, speech, and language processing* 22(10):1533–1545. 2.3

[2] Adomavicius, Gediminas and Alexander Tuzhilin. 2005. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *IEEE transactions on knowledge and data engineering* 17(6):734–749. 1.1, 1.2

[3] Aggarwal, Charu C. 2016. Content-based recommender systems. In *Recommender Systems*. Springer pp. 139–166. 1.1

[4] Allen, Micah, Davide Poggiali, Kirstie Whitaker, Tom Rhys Marshall and Rogier A Kievit. 2019. "Raincloud plots: a multi-platform tool for robust data visualization." *Wellcome open research* 4. 4.4

[5] Alvino, Chris and Justin Basilico. 2015. "Learning a Personalized Homepage." https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a. Accessed: 2021-05-03. 3.4.2

[6] Amatriain, Xavier, Alejandro Jaimes, Nuria Oliver and Josep M Pujol. 2011. Data mining methods for recommender systems. In *Recommender systems handbook*. Springer pp. 39–71. 3.3

[7] Bangor, Aaron, Philip Kortum and James Miller. 2009. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale." *J. Usability Studies* 4(3):114–123. 4.4

[8] Bangor, Aaron, Philip T Kortum and James T Miller. 2008. "An empirical evaluation of the system usability scale." *Intl. Journal of Human–Computer Interaction* 24(6):574–594. 3.5.2

[9] Beel, Joeran, Bela Gipp, Stefan Langer and Corinna Breitinger. 2016. "Paper recommender systems: a literature survey." *International Journal on Digital Libraries* 17(4):305–338. 3.2

[10] Blei, David M, Andrew Y Ng and Michael I Jordan. 2003. "Latent dirichlet allocation." *the Journal of machine Learning research* 3:993–1022. 3.5.1, 4.1

[11] Blier, Leonard. 2016. "A brief report of the Heuritech Deep Learning Meetup #5." https://heuritech.wordpress.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/. Accessed: 2021-05-04. (document), 3.1

[12] Bobadilla, J., F. Ortega, A. Hernando and A. GutiéRrez. 2013. "Recommender Systems Survey." *Know.-Based Syst.* 46:109–132.
URL: *https://doi.org/10.1016/j.knosys.2013.03.012* 3.4.3

[13] Bocanegra, Carlos Luis Sanchez, Jose Luis Sevillano Ramos, Carlos Rizo, Anton Civit and Luis Fernandez-Luque. 2017. "HealthRecSys: A semantic content-based recommender system to complement health videos." *BMC medical informatics and decision making* 17(1):1–10. 2.3

[14] Bollen, Dirk, Bart P Knijnenburg, Martijn C Willemsen and Mark Graus. 2010. Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*. pp. 63–70. 2.4

[15] Brooke, John. 1996. "SUS: a "quick and dirty" usability scale." *Usability evaluation in industry* 189. 3.4.3, 3.5.2

[16] Burke, Robin. 2002. "Hybrid recommender systems: Survey and experiments." *User modeling and user-adapted interaction* 12(4):331–370. 1.1, 1.2

[17] Burke, Robin. 2007. *Hybrid Web Recommender Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg pp. 377–408.
URL: *https://doi.org/10.1007/978-3-540-72079-9*$_1$22.1

[18] Caldwell, Ben, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin and Jason White. 2008. "Web content accessibility guidelines (WCAG) 2.0." *WWW Consortium (W3C)* 290. 3.4.2

[19] Canini, Luca, Sergio Benini and Riccardo Leonardi. 2013. "Affective recommendation of movies based on selected connotative features." *Circuits and Systems for Video Technology, IEEE Transactions on* 23(4):636–647. 2.2

[20] Çano, Erion and Maurizio Morisio. 2017. Moodylyrics: A sentiment annotated lyrics dataset. In *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*. pp. 118–124. 1.1, 2.3

[21] Cantador, Iván, Alejandro Bellogín and David Vallet. 2010. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM pp. 237–240. 1.1

[22] Cantador, Iván, Ignacio Fernández-Tobías and Alejandro Bellogín. 2013. Relating personality types with user preferences in multiple entertainment domains. In *CEUR workshop proceedings*. Shlomo Berkovsky. 2.4, 3.5.2

[23] Cantador, Iván, Ioannis Konstas and Joemon M Jose. 2011. "Categorising social tags to improve folksonomy-based recommendations." *Web semantics: science, services and agents on the World Wide Web* 9(1):1–15. 1.1

[24] Chen, Li and Ho Keung Tsoi. 2011. Users' decision behavior in recommender interfaces: Impact of layout design. In *RecSys' 11 Workshop on Human Decision Making in Recommender Systems*. 2.4

[25] Clement, J. N.d. "Global digital population as of January 2021." https://www.statista.com/statistics/617136/digital-population-worldwide. Accessed: 2021-05-03. 3.4.1

[26] Cremonesi, Paolo, Mehdi Elahi and Franca Garzotto. 2017. "User interface patterns in recommendation-empowered content intensive multimedia applications." *Multimedia Tools and Applications* 76(4):5275–5309. 3.4.3

[27] de Gemmis, Marco, Pasquale Lops, Cataldo Musto, Fedelucio Narducci and Giovanni Semeraro. 2015. Semantics-Aware Content-Based Recommender Systems. In *Recommender Systems Handbook*. Springer pp. 119–159. 3.2

[28] De Gemmis, Marco, Pasquale Lops, Giovanni Semeraro and Pierpaolo Basile. 2008. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM pp. 163–170. 1.1

[29] Deldjoo, Y., M. Schedl and M. Elahi. 2019. Movie Genome Recommender: A Novel Recommender System Based on Multimedia Content. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. pp. 1–4. 2.2, 2.4, 2.5, 3.4

[30] Deldjoo, Yashar, Mehdi Elahi, Massimo Quadrana and Paolo Cremonesi. 2018. "Using visual features based on mpeg-7 and deep learning for movie recommendation." *International Journal of Multimedia Information Retrieval* . 1.1, 2.1

[31] Deldjoo, Yashar, Mehdi Elahi, Massimo Quadrana, Paolo Cremonesi and Franca Garzotto. 2015. Toward Effective Movie Recommendations Based on Mise-en-Scène Film Styles. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*. ACM pp. 162–165. 1.1, 3

[32] Deldjoo, Yashar, Mehdi Elahi, P. Cremonesi, Franca Garzotto, Pietro Piazzolla and Massimo Quadrana. 2016*a*. "Content-Based Video Recommendation System Based on Stylistic Visual Features." *Journal on Data Semantics* 5:99–113. 2.2, 2.5

[33] Deldjoo, Yashar, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla and Massimo Quadrana. 2016*b*. "Content-Based Video Recommendation System Based on Stylistic Visual Features." *Journal on Data Semantics* pp. 1–15. 1.1, 2.2, 2.5

[34] Deldjoo, Yashar, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Bogdan Ionescu, Markus Schedl and Paolo Cremonesi. 2018. Audio-Visual Encoding of Multimedia Content for Enhancing Movie Recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys '18 New York, NY, USA: Association for Computing Machinery p. 455–459.
URL: *https://doi.org/10.1145/3240323.3240407* 2.2

[35] Di Noia, Tommaso, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito and Markus Zanker. 2012. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*. ACM pp. 1–8. 1.1

[36] Downs, Julie S., Mandy B. Holbrook, Steve Sheng and Lorrie Faith Cranor. 2010. Are Your Participants Gaming the System? Screening Mechanical Turk Workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10 New York, NY, USA: Association for Computing Machinery p. 2399–2402.
URL: *https://doi.org/10.1145/1753326.1753688* 5.4

[37] Duchi, John, Elad Hazan and Yoram Singer. 2011. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." 12(null):2121–2159. 3.5.1

[38] Ekstrand, Michael D., F. Maxwell Harper, Martijn C. Willemsen and Joseph A. Konstan. 2014. User Perception of Differences in Recommender Algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys '14 New York, NY, USA: Association for Computing Machinery p. 161–168.
URL: *https://doi.org/10.1145/2645710.2645737* 2.4, 3.4, 3.4.3, 3.5.2, 4.4, 5.3

[39] Elahi, Mehdi, Farshad Bakhshandegan Moghaddam, Reza Hosseini, Rimaz Hossein, Nabil El Ioni, Marko Tkalcic, Christoph Trattner and Tammam Tillo. 2021. Recommending Videos in Cold Start With Automatic Visual Tags. 1.1, 2.2, 2.5

[40] Elahi, Mehdi, Reza Hosseini, Mohammad H Rimaz, Farshad B Moghaddam and Christoph Trattner. 2020. Visually-Aware Video Recommendation in the Cold Start. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*. pp. 225–229. 1.1

[41] Elahi, Mehdi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda and Paolo Cremonesi. 2017. Exploring the Semantic Gap for Movie Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM pp. 326–330. 2.2, 2.5, 3.1.1

[42] Filho, R. J. R., J. Wehrmann and R. C. Barros. 2017. Leveraging deep visual features for content-based movie recommender systems. In *2017 International Joint Conference on Neural Networks (IJCNN)*. pp. 604–611. 2.2, 2.5

[43] Gedikli, Fatih and Dietmar Jannach. 2013. "Improving recommendation accuracy based on item-specific tag preferences." *ACM Transactions on Intelligent Systems and Technology (TIST)* 4(1):11. 1.1

[44] Goldberg, David, David Nichols, Brian M. Oki and Douglas Terry. 1992. "Using Collaborative Filtering to Weave an Information Tapestry." *Commun. ACM* 35(12):61–70. **URL:** *https://doi.org/10.1145/138859.138867* 2.1

[45] Gomez-Uribe, Carlos A and Neil Hunt. 2015. "The netflix recommender system: Algorithms, business value, and innovation." *ACM Transactions on Management Information Systems (TMIS)* 6(4):1–19. 3.4.2

[46] Goodfellow, Ian, Yoshua Bengio and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org. 1.1

[47] Gosling, Samuel D, Peter J Rentfrow and William B Swann Jr. 2003. "A very brief measure of the Big-Five personality domains." *Journal of Research in personality* 37(6):504–528. 3.1, 3.5.2

[48] Gossi, Derek and Mehmet H Gunes. 2016. Lyric-based music recommendation. In *Complex networks VII*. Springer pp. 301–310. 1.1, 2.3

[49] Graves, Alex, Abdel-rahman Mohamed and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee pp. 6645–6649. 2.3

[50] Gunawardana, Asela and Guy Shani. 2015. Evaluating recommender systems. In *Recommender systems handbook*. Springer pp. 265–308. 3.4.3

[51] Harper, F Maxwell and Joseph A Konstan. 2016. "The movielens datasets: History and context." *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19. 3.5.1

[52] Hawashin, Bilal, Mohammad Lafi, Tarek Kanan and Ayman Mansour. 2019. "An efficient hybrid similarity measure based on user interests for recommender systems." *Expert Systems* p. e12471. 2.1

[53] He, Ruining and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web*. WWW '16 Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee p. 507–517. **URL:** *https://doi.org/10.1145/2872427.2883037* 1.1

[54] Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen and John T. Riedl. 2004. "Evaluating Collaborative Filtering Recommender Systems." *ACM Trans. Inf. Syst.* 22(1):5–53.
URL: *https://doi.org/10.1145/963770.963772* 3.5.1

[55] Hong, Liang Jie. 2012. "Pairwise Loss (WARP).". Online; accessed 2021-01-21.
URL: *http://www.hongliangjie.com/2012/08/24/weighted-approximately-ranked-pairwise-loss-warp/* 3.3

[56] Jain, Sarika, Anjali Grover, Praveen Singh Thakur and Sourabh Kumar Choudhary. 2015. Trends, problems and solutions of recommender system. In *International conference on computing, communication & automation*. IEEE pp. 955–958. 1.1

[57] Jannach, Dietmar, Markus Zanker, Alexander Felfernig and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction*. Cambridge University Press. 2.1

[58] Jones, Karen Sparck. 1972. "A statistical interpretation of term specificity and its application in retrieval." *Journal of documentation* . 3.2

[59] Kammerer, Yvonne and Peter Gerjets. 2010. How the interface design influences users' spontaneous trustworthiness evaluations of web search results: comparing a list and a grid interface. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. pp. 299–306. 2.4

[60] Kang, W., C. Fang, Z. Wang and J. McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *2017 IEEE International Conference on Data Mining (ICDM)*. pp. 207–216. 1.1

[61] Knijnenburg, Bart P and Martijn C Willemsen. 2015. Evaluating recommender systems with user experiments. In *Recommender Systems Handbook*. Springer pp. 309–352. 2.4

[62] Knijnenburg, Bart P., Martijn C. Willemsen, Zeno Gantner, Hakan Soncu and Chris Newell. 2012. "Explaining the user experience of recommender systems." *User Modeling and User-Adapted Interaction* 22(4-5):441–504. (document), 1.1, 2.4, 2.4, 2.3, 3.4.3

[63] Konstan, Joseph A and John Riedl. 2012. "Recommender systems: from algorithms to user experience." *User modeling and user-adapted interaction* 22(1):101–123. 2.4

[64] Kortum, Philip and Frederick L Oswald. 2018. "The impact of personality on the subjective assessment of usability." *International Journal of Human–Computer Interaction* 34(2):177–186. 2.4

[65] Kula, Maciej. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015.*, ed. Toine Bogers and Marijn Koolen. Vol. 1448 of *CEUR Workshop Proceedings* CEUR-WS.org pp. 14–21.
URL: *http://ceur-ws.org/Vol-1448/paper4.pdf* 3.3

[66] Laurier, Cyril, Jens Grivolla and Perfecto Herrera. 2008. Multimodal music mood classification using audio and lyrics. In *2008 Seventh International Conference on Machine Learning and Applications*. IEEE pp. 688–693. 2.3

[67] Lee, Joonseok, Sami Abu-El-Haija, Balakrishnan Varadarajan and Apostol (Paul) Natsev. 2018. Collaborative Deep Metric Learning for Video Understanding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18 New York, NY, USA: Association for Computing Machinery p. 481–490.
URL: *https://doi.org/10.1145/3219819.3219856* 2.2

[68] Lehinevych, Taras, Nikolaos Kokkinis-Ntrenis, Giorgos Siantikos, A Seza Dogruöz, Theodoros Giannakopoulos and Stasinos Konstantopoulos. 2014. Discovering similarities for content-based recommendation and browsing in multimedia collections. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*. IEEE pp. 237–243. 2.2

[69] Lewis, James R. 2018. "The System Usability Scale: Past, Present, and Future." *International Journal of Human–Computer Interaction* 34(7):577–590.
URL: *https://doi.org/10.1080/10447318.2018.1455307* 3.5.2

[70] Lewis, James R and Jeff Sauro. 2009. The factor structure of the system usability scale. In *International conference on human centered design*. Springer pp. 94–103. 3.5.2

[71] Li, Y., H. Wang, H. Liu and B. Chen. 2017. A study on content-based video recommendation. In *2017 IEEE International Conference on Image Processing (ICIP)*. pp. 4581–4585. 2.2

[72] Lim, Daryl, Gert Lanckriet and Brian McFee. 2013. Robust structural metric learning. In *International conference on machine learning*. PMLR pp. 615–623. 1.1, 2.3

[73] Ling, Charles X., Jin Huang and Harry Zhang. 2003. AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In *Proceedings of the 16th Canadian Society for Computational Studies of Intelligence Conference on Advances in Artificial Intelligence*. AI'03 Berlin, Heidelberg: Springer-Verlag p. 329–341. 3.5.1

[74] Lops, Pasquale, Marco De Gemmis and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer pp. 73–105. 1.1, 2.1

[75] Lops, Pasquale, Marco De Gemmis, Giovanni Semeraro, Cataldo Musto and Fedelucio Narducci. 2013. "Content-based and collaborative techniques for tag recommendation: an empirical evaluation." *Journal of Intelligent Information Systems* 40(1):41–61. 1.1

[76] Lü, Linyuan, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang and Tao Zhou. 2012. "Recommender systems." *Physics Reports* 519(1):1–49. Recommender Systems. **URL:** *https://www.sciencedirect.com/science/article/pii/S0370157312000828* 3.5.1

[77] Martins, Eder F, Fabiano M Belém, Jussara M Almeida and Marcos A Gonçalves. 2016. "On cold start for associative tag recommendation." *Journal of the Association for Information Science and Technology* 67(1):83–105. 2.1

[78] McAuley, Julian, Christopher Targett, Qinfeng Shi and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '15 New York, NY, USA: Association for Computing Machinery p. 43–52. **URL:** *https://doi.org/10.1145/2766462.2767755* 1.1

[79] McCrae, Robert R and Oliver P John. 1992. "An introduction to the five-factor model and its applications." *Journal of personality* 60(2):175–215. 2.1

[80] McFee, Brian and Gert RG Lanckriet. 2012. Hypergraph Models of Playlist Dialects. In *ISMIR*. Vol. 12 Citeseer pp. 343–348. 1.1, 2.3

[81] McNee, Sean M, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan and John Riedl. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. pp. 116–125. 2.4

[82] Mihalcea, Rada and Carlo Strapparava. 2012. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pp. 590–599. 2.3

[83] Milicevic, Aleksandra Klasnja, Alexandros Nanopoulos and Mirjana Ivanovic. 2010. "Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions." *Artificial Intelligence Review* 33(3):187–209. 1.1

[84] Moghaddam, Farshad B, Mehdi Elahi, Reza Hosseini, Christoph Trattner and Marko Tkalčič. 2019*a*. Predicting movie popularity and ratings with visual features. In *2019*

14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). IEEE pp. 1–6. (document), 1.1, 2.2, 2.2, 2.5, 3.1.1

[85] Moghaddam, Farshad B., Mehdi Elahi, Reza Hosseini, Christoph Trattner and Marko Tkalcic. 2019*b*. Predicting Movie Popularity and Ratings with Visual Features. In *2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2019*. Number May. 2.2

[86] Moghaddam, Farshad Bakhshandegan and Mehdi Elahi. 2019. "Cold Start Solutions For Recommendation Systems Music recommender systems.".
URL: *https://www.researchgate.net/publication/332511384* 2.1, 3.4.3, 3.5.2

[87] Open-source and Google. 2020. "Keras.". Online; accessed 2021-01-21.
URL: *https://github.com/tensorflow/tensorflow/tree/master/tensorflow/python/keras* 3.1.1

[88] opensubtitles.org. 2020. "OpenSubtitles.". Online; accessed 2020-11-01.
URL: *https://trac.opensubtitles.org/projects/opensubtitles* 3.1.2

[89] Oppenheimer, Daniel, Tom Meyvis and Nicolas Davidenko. 2009. "Instructional Manipulation Checks: Detecting Satisficing to Increase Statistical Power." *Journal of Experimental Social Psychology* 45:867–872. (document), 3.6, 3.5.2

[90] Orne, M. 1962. "On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications." *American Psychologist* 17:776–783. 5.4

[91] Palan, Stefan and Christian Schitter. 2018. "Prolific.ac—A subject pool for online experiments." *Journal of Behavioral and Experimental Finance* 17:22–27.
URL: *https://www.sciencedirect.com/science/article/pii/S2214635017300989* 5.4

[92] Pan, Weike, Hao Zhong, Congfu Xu and Zhong Ming. 2015. "Adaptive Bayesian Personalized Ranking for Heterogeneous Implicit Feedbacks." *Know.-Based Syst.* 73(1):173–180.
URL: *https://doi.org/10.1016/j.knosys.2014.09.013* 3.3

[93] Parra, Denis, Alexandros Karatzoglou, Xavier Amatriain and Idil Yavuz. 2011. "Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping." *Proceedings of the CARS-2011* 5. 3.3

[94] Peer, Eyal, Laura Brandimarte, Sonam Samat and Alessandro Acquisti. 2017. "Beyond the Turk: Alternative platforms for crowdsourcing behavioral research." *Journal of Experimental Social Psychology* 70:153–163.
URL: *https://www.sciencedirect.com/science/article/pii/S0022103116303201* 5.4

[95] Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz et al. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. Number CONF IEEE Signal Processing Society. 2.3

[96] Pu, Pearl, Li Chen and Rong Hu. 2011. "A user-centric evaluation framework for recommender systems." *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems* pp. 157–164. 2.4, 3.5.2

[97] Rawlings, David and Vera Ciancarelli. 1997. "Music preference and the five-factor model of the NEO Personality Inventory." *Psychology of Music* 25(2):120–132. 2.4, 3.5.2

[98] Renckes, Sahin, Huseyin Polat and Yusuf Oysal. 2012. "A new hybrid recommendation algorithm with privacy." *Expert Systems* 29(1):39–55. 2.1

[99] Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press pp. 452–461. 3.3, 3.3

[100] Rentfrow, Peter J, Lewis R Goldberg and Ran Zilca. 2011. "Listening, watching, and reading: The structure and correlates of entertainment preferences." *Journal of personality* 79(2):223–258. 2.4, 3.5.2

[101] Rentfrow, Peter J and Samuel D Gosling. 2003. "The do re mi's of everyday life: the structure and personality correlates of music preferences." *Journal of personality and social psychology* 84(6):1236. 2.4, 3.5.2

[102] Resnick, Paul and Hal R Varian. 1997*a*. "Recommender systems." *Communications of the ACM* 40(3):56–58. 1.1

[103] Resnick, Paul and Hal R. Varian. 1997*b*. "Recommender systems." *Commun. ACM* 40(3):56–58. 2.1

[104] Ricci, Francesco, Lior Rokach and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, ed. Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor. Springer Verlag pp. 1–35. (document), 2.1, 2.1

[105] Ricci, Francesco, Lior Rokach and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. Springer US pp. 1–34. 1.2

[106] Rimaz, Mohammad Hossein, Mehdi Elahi, Farshad Bakhshandegan Moghadam, Christoph Trattner, Reza Hosseini and Marko Tkalčič. 2019. "Exploring the power of visual features for the recommendation of movies." *ACM UMAP 2019 - Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization* (June):303–308. 2.2

[107] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. 2015. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision (IJCV)* 115(3):211–252. 3.1.1

[108] Sauro, Jeff. 2011. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC. 4.4, 4.4

[109] Sauro, Jeff and James R. Lewis. 2016. Chapter 8 - Standardized usability questionnaires. In *Quantifying the User Experience (Second Edition)*, ed. Jeff Sauro and James R. Lewis. Second edition ed. Boston: Morgan Kaufmann pp. 185–248.
URL: *https://www.sciencedirect.com/science/article/pii/B9780128023082000084* 3.5.2, 4.4

[110] Schedl, Markus, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo and Mehdi Elahi. 2018. "Current Challenges and Visions in Music Recommender Systems Research." *International Journal of Multimedia Information Retrieval* 7. 3.5.1, 3.5.1

[111] Shani, Guy and Asela Gunawardana. 2011. *Evaluating Recommendation Systems*. Boston, MA: Springer US pp. 257–297.
URL: *https://doi.org/10.1007/978-0-387-85820-3*$_8$ 3.4.2, 3.4.3, 5.4

[112] Shepitsen, Andriy, Jonathan Gemmell, Bamshad Mobasher and Robin Burke. 2008. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM pp. 259–266. 1.1

[113] Shi, Yue, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver and Alan Hanjalic. 2012. CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys '12 New York, NY, USA: Association for Computing Machinery p. 139–146.
URL: *https://doi.org/10.1145/2365952.2365981* 3.5.1

[114] Shi, Yue, Martha Larson and Alan Hanjalic. 2010. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Proceedings of the workshop on context-aware movie recommendation*. pp. 34–40. 2.3

[115] Simonyan, Karen and Andrew Zisserman. 2015. “Very deep convolutional networks for large-scale image recognition.” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* pp. 1–14. (document), 3.1, 3.1.1

[116] Statcounter. N.d. “Desktop vs Mobile vs Tablet Market Share Worldwide.” https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide. Accessed: 2021-05-03. 3.4.1

[117] Steele-Johnson, Debra, Russell Beauregard, Paul Hoover and Aaron Schmidt. 2000. “Goal orientation and task demand effects on motivation, affect, and performance.” *The Journal of applied psychology* 85:724–38. 5.4

[118] Sulthana, A. Razia, Maulika Gupta, Shruthi Subramanian and Sakina Mirza. 2020. “Improvising the performance of image-based recommendation system using convolution neural networks and deep learning.” *Soft Computing* 0.
URL: *https://doi.org/10.1007/s00500-020-04803-0* 2.2

[119] Tkalcic, Marko and Li Chen. 2015. Personality and recommender systems. In *Recommender systems handbook.* Springer pp. 715–739. 2.4

[120] Torres, Roberto, Sean M McNee, Mara Abel, Joseph A Konstan and John Riedl. 2004. Enhancing digital libraries with TechLens+. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries.* pp. 228–236. 2.4

[121] Van der Maaten, Laurens and Geoffrey Hinton. 2008. “Visualizing data using t-SNE.” *Journal of machine learning research* 9(11). 3.5.1, 4.1

[122] Vijayarani, S, Ms J Ilamathi, Ms Nithya et al. 2015. “Preprocessing techniques for text mining-an overview.” *International Journal of Computer Science & Communication Networks* 5(1):7–16. 3.2

[123] Voorhees, Ellen. 2000. “The TREC-8 Question Answering Track Report.”.
URL: *https://tsapps.nist.gov/publication/*$get_pdf.cfm?pub_id = 151495$ 3.5.1

[124] Wang, Lichuan, Xianyi Zeng, Ludovic Koehl and Yan Chen. 2015. “Intelligent Fashion Recommender System: Fuzzy Logic in Personalized Garment Design.” *IEEE Trans. Human-Machine Systems* 45(1):95–109. 1.1

[125] Wang, Yu, ChunXiao Xing and Lizhu Zhou. 2006. “Video semantic models: survey and evaluation.” *Int. J. Comput. Sci. Netw. Security* 6:10–20. 2.1

[126] Weston, Jason, Samy Bengio and Nicolas Usunier. 2011. WSABIE: Scaling up to Large Vocabulary Image Annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three.* IJCAI’11 AAAI Press p. 2764–2770. 3.3

[127] Wold, Svante, Kim Esbensen and Paul Geladi. 1987. “Principal component analysis.” *Chemometrics and intelligent laboratory systems* 2(1-3):37–52. 3.5.1, 4.1

[128] Yang, Bo, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang and Mingjing Li. 2007. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM pp. 73–80. 2.2

[129] Zhao, Xiaojian, Guangda Li, Meng Wang, Jin Yuan, Zheng-Jun Zha, Zhoujun Li and Tat-Seng Chua. 2011. Integrating rich information for video recommendation with multi-task rank aggregation. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM pp. 1521–1524. 2.2

[130] Zhou, Tao, Linyuan Lü and Yi-Cheng Zhang. 2009. “Predicting Missing Links via Local Information.” *The European Physical Journal B - Condensed Matter and Complex Systems* 71:623–630. 3.5.1

[131] Ziegler, Cai-Nicolas, Sean M. McNee, Joseph A. Konstan and Georg Lausen. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web*. WWW ’05 New York, NY, USA: Association for Computing Machinery p. 22–32.
URL: *https://doi.org/10.1145/1060745.1060754* 3.5.2

Appendix A: User Study Statistics

Visualizations of statistics from the user study that are not included in the Results chapter are included in this appendix.

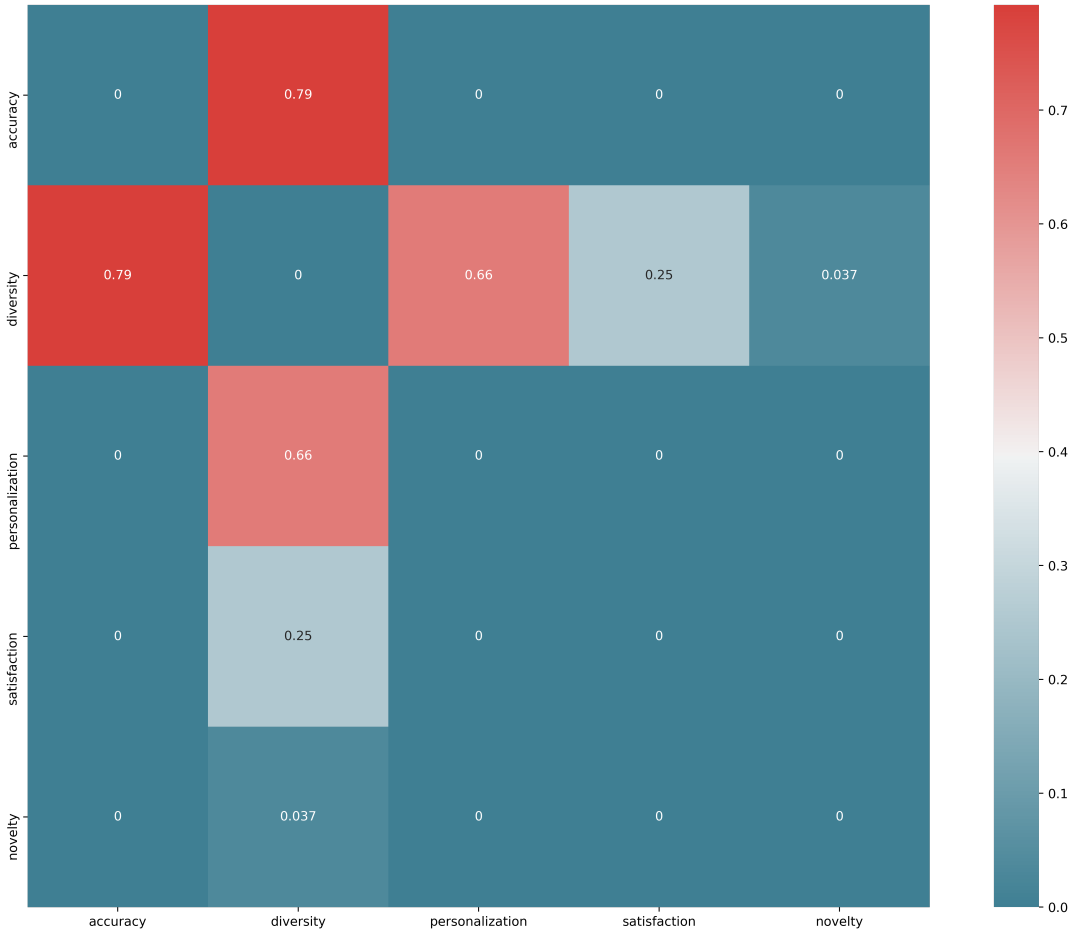
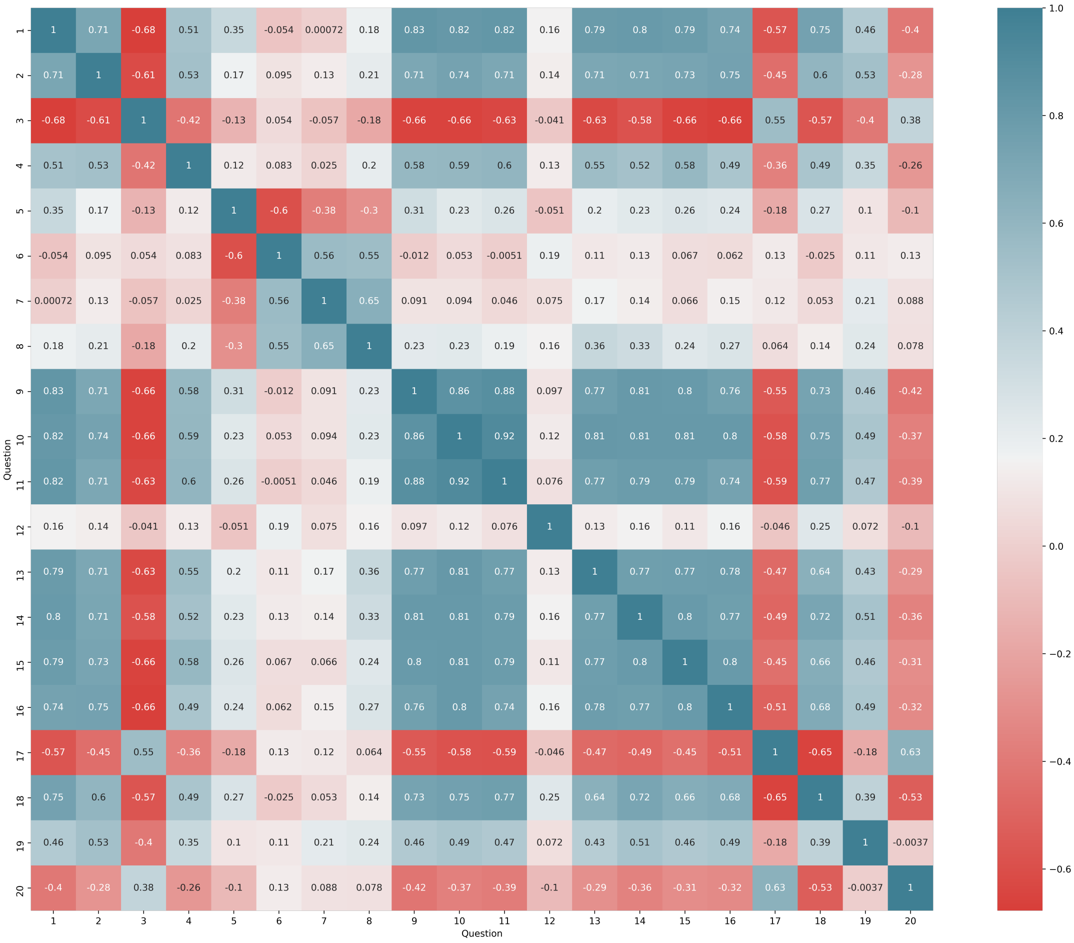
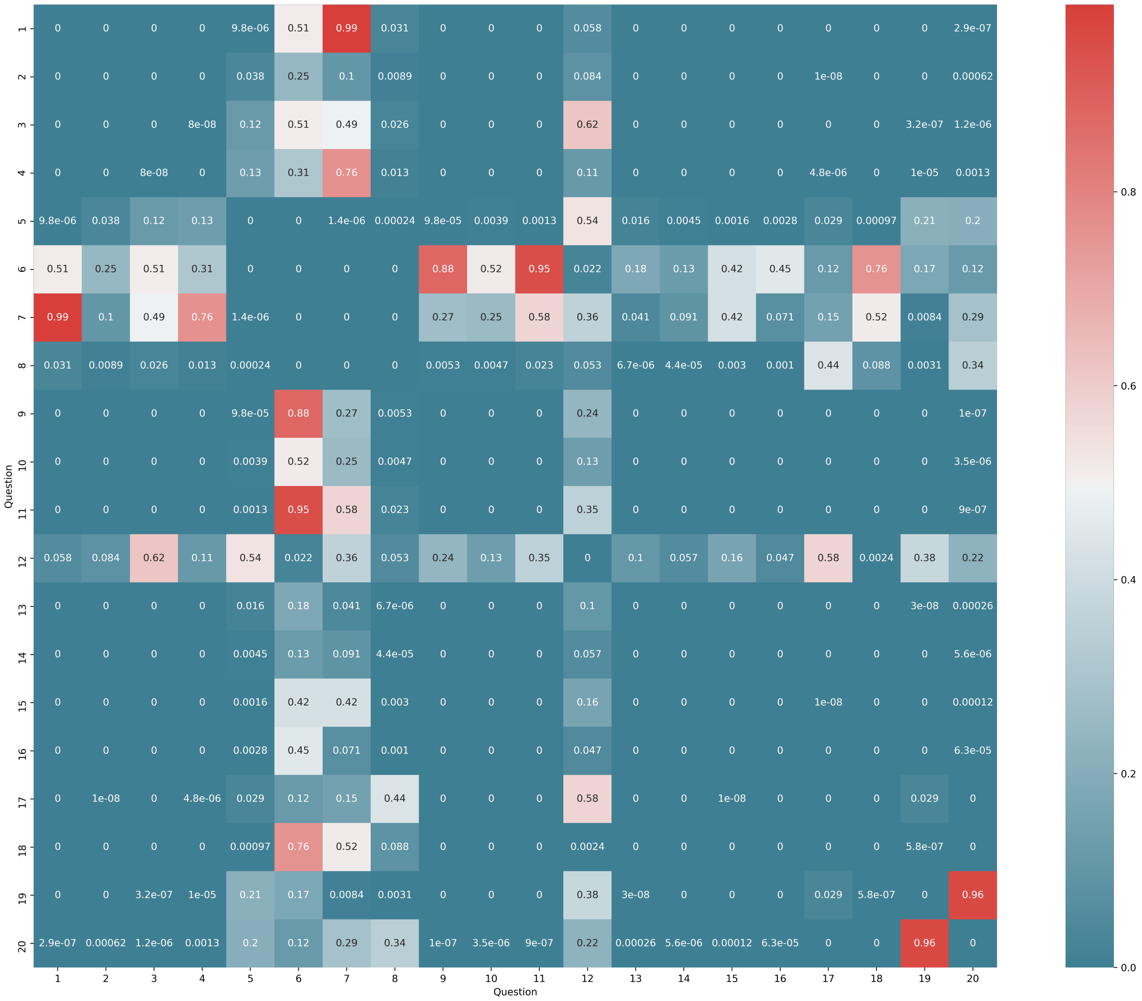


Figure 1: The p-values between the different recommendation quality factors of the user study.



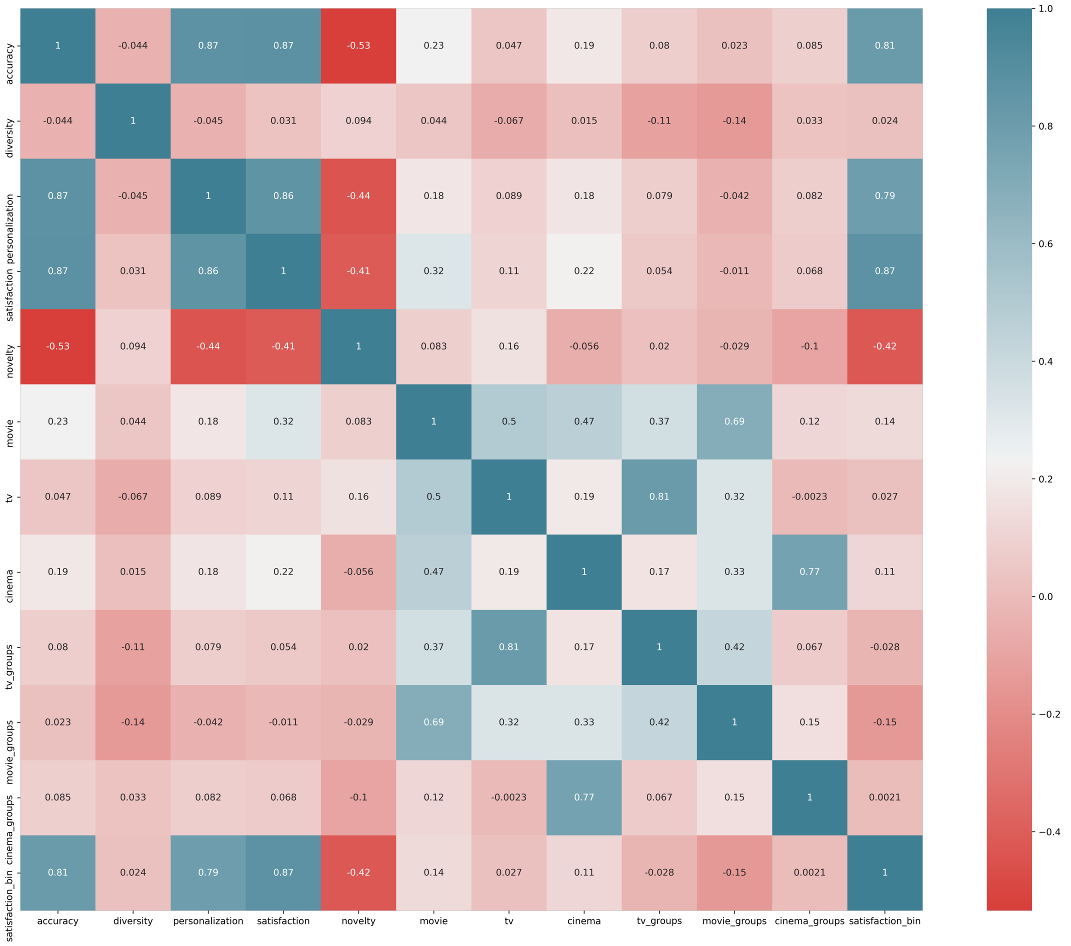
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0.71	-0.68	0.51	0.35	-0.054	0.00072	0.18	0.83	0.82	0.82	0.16	0.79	0.8	0.79	0.74	-0.57	0.75	0.46	-0.4
2	0.71	1	-0.61	0.53	0.17	0.095	0.13	0.21	0.71	0.74	0.71	0.14	0.71	0.71	0.73	0.75	-0.45	0.6	0.53	-0.28
3	-0.68	-0.61	1	-0.42	-0.13	0.054	-0.057	-0.18	-0.66	-0.66	-0.63	-0.041	-0.63	-0.58	-0.66	-0.66	0.55	-0.57	-0.4	0.38
4	0.51	0.53	-0.42	1	0.12	0.083	0.025	0.2	0.58	0.59	0.6	0.13	0.55	0.52	0.58	0.49	-0.36	0.49	0.35	-0.26
5	0.35	0.17	-0.13	0.12	1	-0.6	-0.38	-0.3	0.31	0.23	0.26	-0.051	0.2	0.23	0.26	0.24	-0.18	0.27	0.1	-0.1
6	-0.054	0.095	0.054	0.083	-0.6	1	0.56	0.55	-0.012	0.053	-0.0051	0.19	0.11	0.13	0.067	0.062	0.13	-0.025	0.11	0.13
7	0.00072	0.13	-0.057	0.025	-0.38	0.56	1	0.65	0.091	0.094	0.046	0.075	0.17	0.14	0.066	0.15	0.12	0.053	0.21	0.088
8	0.18	0.21	-0.18	0.2	-0.3	0.55	0.65	1	0.23	0.23	0.19	0.16	0.36	0.33	0.24	0.27	0.064	0.14	0.24	0.078
9	0.83	0.71	-0.66	0.58	0.31	-0.012	0.091	0.23	1	0.86	0.88	0.097	0.77	0.81	0.8	0.76	-0.55	0.73	0.46	-0.42
10	0.82	0.74	-0.66	0.59	0.23	0.053	0.094	0.23	0.86	1	0.92	0.12	0.81	0.81	0.81	0.8	-0.58	0.75	0.49	-0.37
11	0.82	0.71	-0.63	0.6	0.26	-0.0051	0.046	0.19	0.88	0.92	1	0.076	0.77	0.79	0.79	0.74	-0.59	0.77	0.47	-0.39
12	0.16	0.14	-0.041	0.13	-0.051	0.19	0.075	0.16	0.097	0.12	0.076	1	0.13	0.16	0.11	0.16	-0.046	0.25	0.072	-0.1
13	0.79	0.71	-0.63	0.55	0.2	0.11	0.17	0.36	0.77	0.81	0.77	0.13	1	0.77	0.77	0.78	-0.47	0.64	0.43	-0.29
14	0.8	0.71	-0.58	0.52	0.23	0.13	0.14	0.33	0.81	0.81	0.79	0.16	0.77	1	0.8	0.77	-0.49	0.72	0.51	-0.36
15	0.79	0.73	-0.66	0.58	0.26	0.067	0.066	0.24	0.8	0.81	0.79	0.11	0.77	0.8	1	0.8	-0.45	0.66	0.46	-0.31
16	0.74	0.75	-0.66	0.49	0.24	0.062	0.15	0.27	0.76	0.8	0.74	0.16	0.78	0.77	0.8	1	-0.51	0.68	0.49	-0.32
17	-0.57	-0.45	0.55	-0.36	-0.18	0.13	0.12	0.064	-0.55	-0.58	-0.59	-0.046	-0.47	-0.49	-0.45	-0.51	1	-0.65	-0.18	0.63
18	0.75	0.6	-0.57	0.49	0.27	-0.025	0.053	0.14	0.73	0.75	0.77	0.25	0.64	0.72	0.66	0.68	-0.65	1	0.39	-0.53
19	0.46	0.53	-0.4	0.35	0.1	0.11	0.21	0.24	0.46	0.49	0.47	0.072	0.43	0.51	0.46	0.49	-0.18	0.39	1	-0.0037
20	-0.4	-0.28	0.38	-0.26	-0.1	0.13	0.088	0.078	-0.42	-0.37	-0.39	-0.1	-0.29	-0.36	-0.31	-0.32	0.63	-0.53	-0.0037	1

Figure 2: The correlations between the each of the questions in the recommendation quality evaluation of the user study.



Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	9.8e-06	0.51	0.99	0.031	0	0	0	0.058	0	0	0	0	0	0	0	2.9e-07
2	0	0	0	0	0.038	0.25	0.1	0.0089	0	0	0	0.084	0	0	0	0	1e-08	0	0	0.00062
3	0	0	0	8e-08	0.12	0.51	0.49	0.026	0	0	0	0.62	0	0	0	0	0	0	3.2e-07	1.2e-06
4	0	0	8e-08	0	0.13	0.31	0.76	0.013	0	0	0	0.11	0	0	0	0	4.8e-06	0	1e-05	0.0013
5	9.8e-06	0.038	0.12	0.13	0	0	1.4e-06	0.00024	9.8e-05	0.0039	0.0013	0.54	0.016	0.0045	0.0016	0.0028	0.029	0.00097	0.21	0.2
6	0.51	0.25	0.51	0.31	0	0	0	0	0.88	0.52	0.95	0.022	0.18	0.13	0.42	0.45	0.12	0.76	0.17	0.12
7	0.99	0.1	0.49	0.76	1.4e-06	0	0	0	0.27	0.25	0.58	0.36	0.041	0.091	0.42	0.071	0.15	0.52	0.0084	0.29
8	0.031	0.0089	0.026	0.013	0.00024	0	0	0	0.0053	0.0047	0.023	0.053	6.7e-06	4.4e-05	0.003	0.001	0.44	0.088	0.0031	0.34
9	0	0	0	0	9.8e-05	0.88	0.27	0.0053	0	0	0	0.24	0	0	0	0	0	0	0	1e-07
10	0	0	0	0	0.0039	0.52	0.25	0.0047	0	0	0	0.13	0	0	0	0	0	0	0	3.5e-06
11	0	0	0	0	0.0013	0.95	0.58	0.023	0	0	0	0.35	0	0	0	0	0	0	0	9e-07
12	0.058	0.084	0.62	0.11	0.54	0.022	0.36	0.053	0.24	0.13	0.35	0	0.1	0.057	0.16	0.047	0.58	0.0024	0.38	0.22
13	0	0	0	0	0.016	0.18	0.041	6.7e-06	0	0	0	0.1	0	0	0	0	0	0	3e-08	0.00026
14	0	0	0	0	0.0045	0.13	0.091	4.4e-05	0	0	0	0.057	0	0	0	0	0	0	0	5.6e-06
15	0	0	0	0	0.0016	0.42	0.42	0.003	0	0	0	0.16	0	0	0	0	1e-08	0	0	0.00012
16	0	0	0	0	0.0028	0.45	0.071	0.001	0	0	0	0.047	0	0	0	0	0	0	0	6.3e-05
17	0	1e-08	0	4.8e-06	0.029	0.12	0.15	0.44	0	0	0	0.58	0	0	1e-08	0	0	0	0.029	0
18	0	0	0	0	0.00097	0.76	0.52	0.088	0	0	0	0.0024	0	0	0	0	0	0	5.8e-07	0
19	0	0	3.2e-07	1e-05	0.21	0.17	0.0084	0.0031	0	0	0	0.38	3e-08	0	0	0	0.029	5.8e-07	0	0.96
20	2.9e-07	0.00062	1.2e-06	0.0013	0.2	0.12	0.29	0.34	1e-07	3.5e-06	9e-07	0.22	0.00026	5.6e-06	0.00012	6.3e-05	0	0	0.96	0

Figure 3: The p-values between the each of the questions in the recommendation quality evaluation of the user study.



	accuracy	diversity	personalization	satisfaction	novelty	movie	tv	cinema	tv_groups	movie_groups	cinema_groups	satisfaction_bin
accuracy	1	-0.044	0.87	0.87	-0.53	0.23	0.047	0.19	0.08	0.023	0.085	0.81
diversity	-0.044	1	-0.045	0.031	0.094	0.044	-0.067	0.015	-0.11	-0.14	0.033	0.024
personalization	0.87	-0.045	1	0.86	-0.44	0.18	0.089	0.18	0.079	-0.042	0.082	0.79
satisfaction	0.87	0.031	0.86	1	-0.41	0.32	0.11	0.22	0.054	-0.011	0.068	0.87
novelty	-0.53	0.094	-0.44	-0.41	1	0.083	0.16	-0.056	0.02	-0.029	-0.1	-0.42
movie	0.23	0.044	0.18	0.32	0.083	1	0.5	0.47	0.37	0.69	0.12	0.14
tv	0.047	-0.067	0.089	0.11	0.16	0.5	1	0.19	0.81	0.32	-0.0023	0.027
cinema	0.19	0.015	0.18	0.22	-0.056	0.47	0.19	1	0.17	0.33	0.77	0.11
tv_groups	0.08	-0.11	0.079	0.054	0.02	0.37	0.81	0.17	1	0.42	0.067	-0.028
movie_groups	0.023	-0.14	-0.042	-0.011	-0.029	0.69	0.32	0.33	0.42	1	0.15	-0.15
cinema_groups	0.085	0.033	0.082	0.068	-0.1	0.12	-0.0023	0.77	0.067	0.15	1	0.0021
satisfaction_bin	0.81	0.024	0.79	0.87	-0.42	0.14	0.027	0.11	-0.028	-0.15	0.0021	1

Figure 4: The correlations between the different recommendation quality factors of the user study and media consumption habits of the participants.

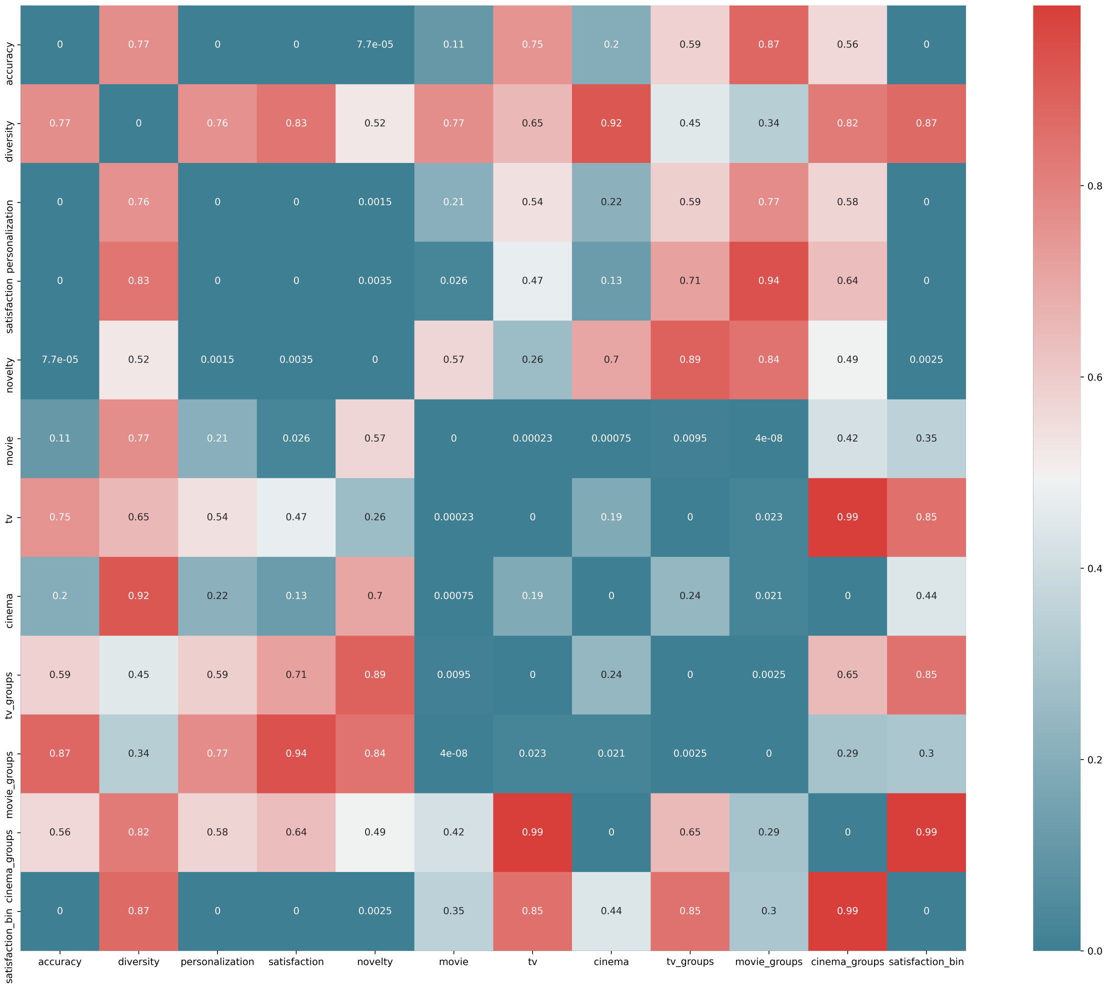


Figure 5: The p-values between the different recommendation quality factors of the user study and media consumption habits of the participants.

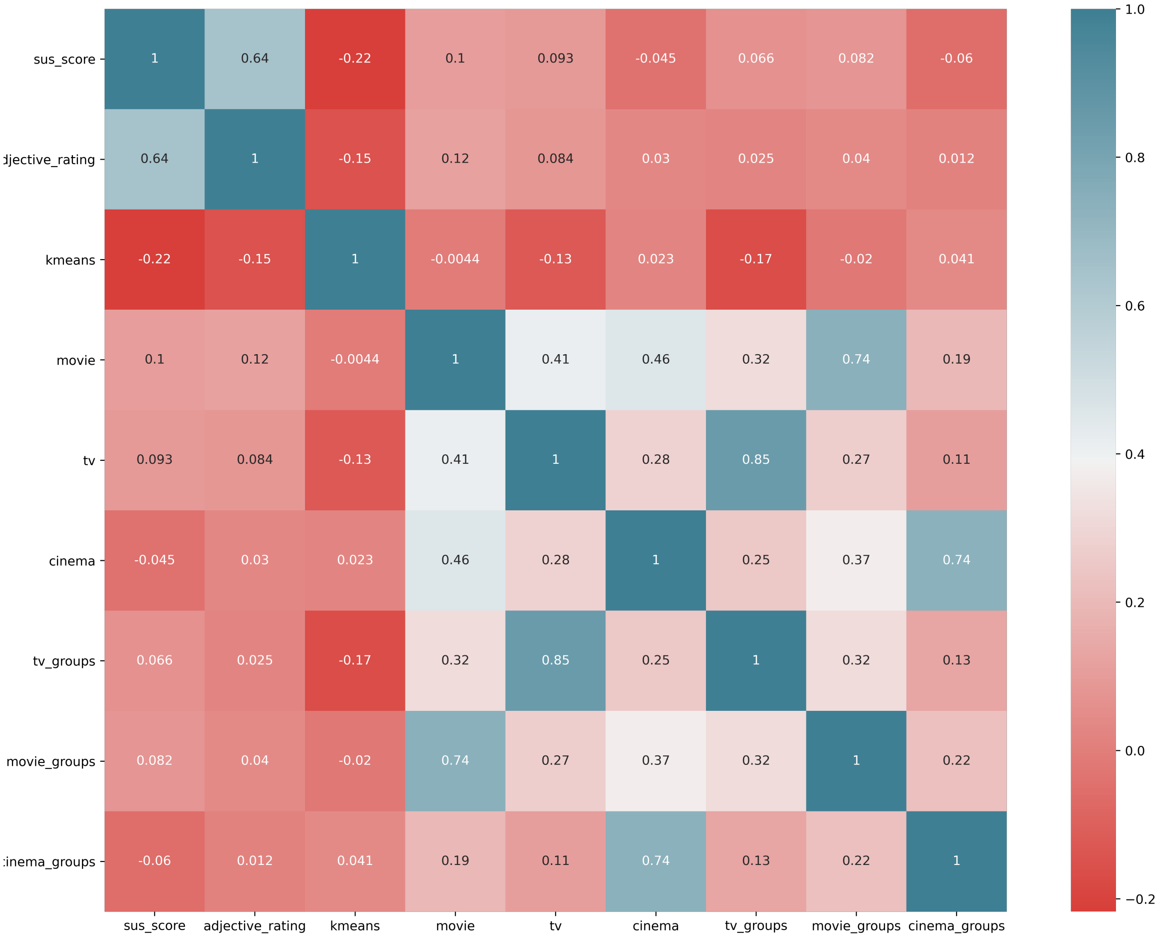


Figure 6: The correlations between the SUS responses and media consumption habits of the participants.

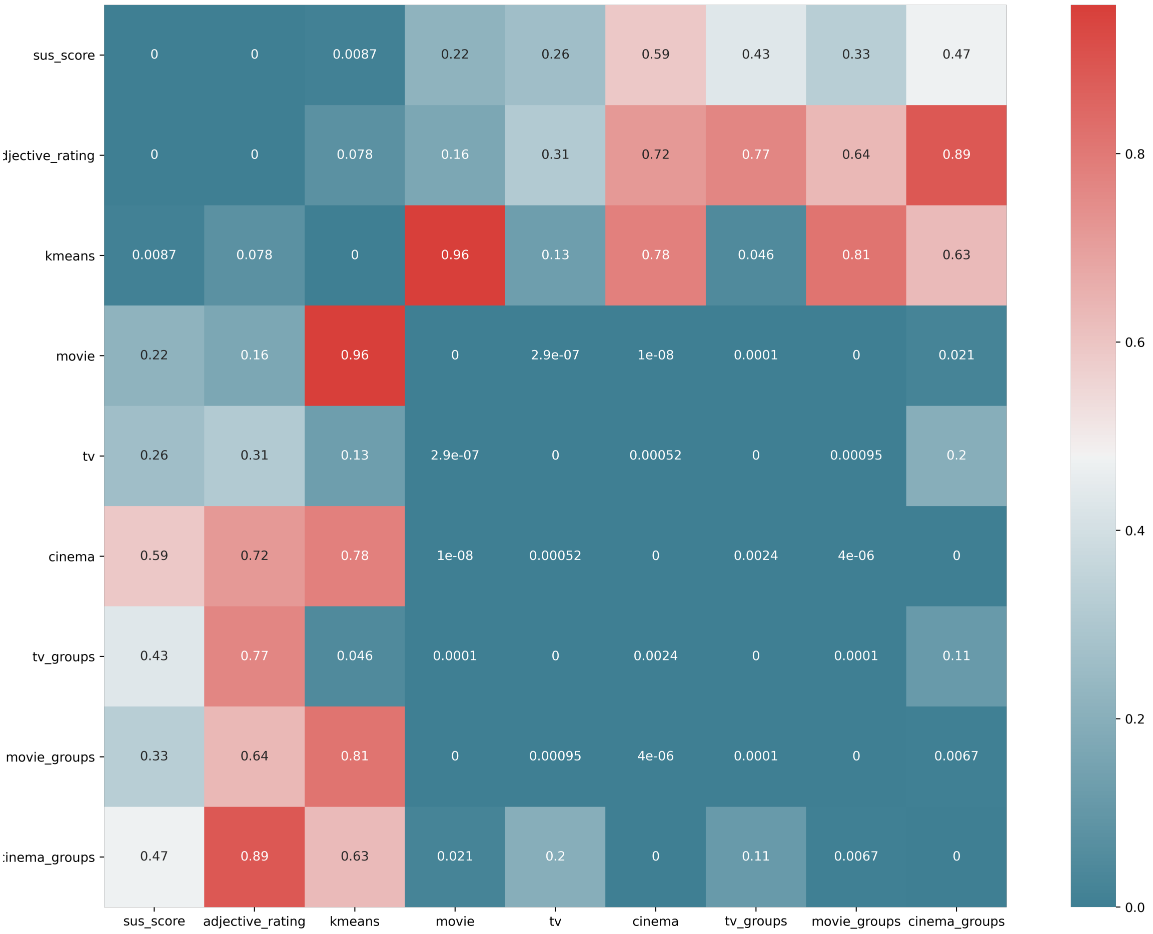


Figure 7: The p-values between the SUS responses and media consumption habits of the participants.